Scheduling cross-docking operations : Integration of operational uncertainties and resource capacities

Anne-Laure Ladier

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Génie Industriel**

Arrêté ministériel : 7 août 2006

Présentée par

Anne-Laure LADIER

Thèse dirigée par **Gülgün ALPAN**

préparée au sein du **Laboratoire G-SCOP**
dans **l'École Doctorale I-MEP^2**

Planification des opérations de cross-docking

Prise en compte des incertitudes opérationnelles et de la capacité des ressources internes

Thèse soutenue publiquement le **21 novembre 2014**,
devant le jury composé de :

Mme Luce BROTCORNE
Chargée de Recherche, HDR, INRIA Lille Nord Europe, Rapporteur

M. Pierre BAPTISTE
Professeur, École Polytechnique de Montréal, Rapporteur

M. Damien TRENTESAUX
Professeur, Université de Valenciennes et du Hainaut-Cambrésis, Président

M. Allen G. GREENWOOD
Professeur, Mississippi State University, Examinateur

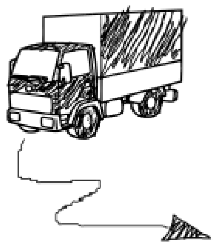
Mme Gülgün ALPAN
Maître de conférences, HDR, Grenoble INP, Directrice de thèse

Anne-Laure Ladier: *Planification des opérations de cross-docking – Prise en compte des incertitudes opérationnelles et de la capacité des ressources internes,* Thèse de doctorat, 21 novembre 2014

SCHEDULING CROSS-DOCKING OPERATIONS

Integration of operational uncertainties and resource capacities

Anne-Laure Ladier



It is good to have an end to journey toward;
but it is the journey that matters, in the end.

— Ernest Hemingway

ACKNOWLEDGEMENTS

Me voici au bout de ce travail de thèse, au bout du voyage ; et j'ai de nombreux compagnons de route à remercier.

Tout commence par Gülgün, bien sûr. Sans ta persévérance et ta patience (devrais-je ajouter ton insistance?) pour me convaincre de commencer cette thèse, l'idée ne m'aurait sans doute pas plus qu'effleurée. Et je n'ai accepté que parce que c'était toi, bien sûr – pas parce que tu avais déjà détaillé le plan de "la thèse d'Anne-Laure" à ton jury d'HDR quand que je n'étais encore sûre de rien. Merci pour ta patience, tes conseils, ta propension à toujours tout trouver *excellent*, ta constance à retirer tous les points d'exclamation de mes écrits (note que sur ce paragraphe j'ai fait un effort), et ta présence infaillible malgré un emploi du temps qui s'est un peu chargé au cours de cette dernière année. Nous allons continuer à travailler ensemble cette année, et j'en suis vraiment ravie et fière.

Je tiens également à remercier chaleureusement Luce Brotcorne et Pierre Baptiste pour m'avoir fait l'honneur d'accepter de rapporter ce travail de thèse. Merci pour l'intérêt que vous avez porté à mon travail, la précision de vos rapports et vos remarques judicieuses. Je remercie également Damien Trentesaux pour avoir présidé mon jury de thèse.

A very special thank you goes to Allen G. Greenwood. For being here during my defense – I believe it was a nice conclusion of all the work done together – but especially for your very warm welcome during my stays in Mississippi State University. You organized everything to make me feel at home right away, patiently corrected my English, and always dedicated time for us to work together. For all this, thank you once again.

A big thank you to my proofreaders : Mamie & Daddy, Hadrien, Philou, Marie-Amélie, Halston and Mark, who all spent time and dedicated a lot of attention to this manuscript.

Ces trois années à G-SCOP m'ont donné l'occasion de passer de l'autre côté du miroir en travaillant avec ceux qui étaient auparavant mes enseignants. Continuer à apprendre à vos côtés à compensé le désarroi de me faire soudainement appeler Madame. Je remercie particulièrement Hadrien Cambazard, Nicolas Catusse, Olivier Briant, Bernard Penz, Maxime Ogier, Fabien Mangione, Pierre Lemaire et tous les autres avec qui j'ai eu l'occasion de travailler dans le cadre de mes missions d'enseignement à l'école de Génie industriel. Si je me suis découvert un véritable goût pour l'enseignement, c'est en grande partie grâce à vous – et aussi aux étudiants de Génie industriel bien sûr.

Bien que le compliment soit devenu bateau à force de l'entendre soutenance après soutenance, je tiens à remercier une fois de plus les membres de G-SCOP dans leur ensemble pour l'ambiance exceptionnelle qui règne au laboratoire. Mais avec une pensée spéciale pour les derniers joueurs de Bang! et les non-joueurs de coinche, les amateurs de yoga ainsi que les bureaux de l'A-DOC passés, présents et futurs – et bien sûr en particulier Maud et Michaël, respectivement trésorière et secrétaire de choc. Maud cumule en plus le fait d'avoir été une cobureau exceptionnelle pendant ces trois ans et pour encore quelques mois de rab'! Bureau que Karim et Julien ont très avantageusement complété pendant cette dernière année. Merci pour tous ces moments passés ensemble. Merci à tous qui êtes passés de collègues à amis.

Merci aux amis "d'avant", Boris et Philou pour nos réunions hebdomadaires (ou presque...), Gabi pour les coups de fil interminables à se raconter nos vies (décidemment compliquées), Sastien, Julie et Mimi pour les escapades dans le sud. Thank you to my lovely co-Boardies, Terhi and Ilka, for the time spent working together for ESTIEM Alumni but also for the time spent discoursing about love, life and ice-creams.

Merci enfin à ma famille. Le fait que vous vous vous soyez déplacés en nombre pour ma soutenance m'a beaucoup touchée, bien que ce ne soient pas les docteurs qui manquent dans la famille. Merci à mes parents bien sûr, qui me témoignent depuis toujours une confiance sans faille. Merci à mon petit frère. J'ai enfin une pensée toute spéciale pour Michel, Jean-Michel, Mamie dont j'ai modestement essayé de suivre les traces, et pour Daddy qui a toujours été pour moi un modèle.

Anne-Laure Ladier,
December 2, 2014

CONTENTS

LISTE DES RÉSUMÉS EN FRANÇAIS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AHP Analytical Hierarchy Process

DC Distribution Center

FIFO First In – First Out

GRASP Greedy Randomized Adaptive Search Procedure

IT Information Technology

IP Integer Program

JIT Just-In-Time

LIFO Last In – First Out

LP Linear Program

LTL Less than Truckload

MILP Mixed and Integer Linear Program

UTA Utilités Additives, *i. e.* additive utilities

WMS Warehouse Management System

INTRODUCTION

The past decades have seen a strong tension of the economical context but also a tremendously fast progress in new technologies. In a changing environment industries have to adapt themselves. Outsourcing the production in countries with lower production costs creates new needs in logistics in order to bring back the products near their European consumers. Selling products online means more intense competition, while consumers are becoming used to free shipping and very fast deliveries. In such a context, companies cannot expect to survive without a fast, efficient and reactive supply-chain. Thus logistic companies and logistic departments start turning their attention to practices that used to be considered as reserved for manufacturing only: lean, Just-In-Time (JIT), or continuous performance improvement techniques.

Cross-docking is a JIT logistic technique. In a regular Distribution Center (DC), items received from inbound trucks are unloaded and put away in storage. When an item is needed it is retrieved from storage (possibly through a picking process), packaged and loaded in an outbound truck. In a cross-docking platform or crossdock, however, items transit directly from inbound trucks to outbound trucks: they are unloaded, dispatched, transferred and reloaded in less than 24 hours, with as little intermediate storage as possible. This technique accelerates the flow of goods and eliminates most of the storage costs.

This dissertation focuses on the logistic operations occurring in a crossdock, and the operational decisions to be made to operate the platform efficiently. Many operational questions need to be answered in the course of daily operations; our first goal is to determine which are the most critical at the moment. A review of the literature on the topic is thus carried out in chapter 1 and compared with the practices observed in industry, thanks to a comparison framework. This enables us to map the existing gaps between the research state-of-the-art and the platform managers' needs. From this study we draw the motivation for this dissertation, which is to fill two of the most critical gaps identified:

How to manage delayed trucks without disturbing other ongoing operations?

How to schedule the workers in a way that fits the operations workload?

Chapter 2, chapter 3 and chapter 4 address the first question, while chapter 5 answers the second one. Finally, chapter 6 proposes to combine the two aspects to address both issues together.

INTRODUCTION

Dans un contexte économique en forte tension et face à la progression fulgurante des nouvelles technologies, les industries doivent s'adapter. La délocalisation crée de nouveaux besoins en matière de logistique, afin de ramener les produits vers leurs consommateurs européens. Vendre ses produits en ligne signifie s'exposer à une concurrence plus intense, tandis que les consommateurs s'habituent à une livraison gratuite dans des délais très courts. Dans ce contexte, les entreprises ne peuvent espérer survivre sans une supply-chain rapide, efficiente et réactive. Les entreprises de logistique et les départements logistiques se tournent donc vers des pratiques longtemps considérées comme l'apanage de la production : le lean, le juste-à-temps ou les méthodes d'amélioration continue.

Le cross-docking est une technique logistique de juste-à-temps. Dans une plateforme logistique classique, les produits reçus sont déchargés des camions entrants puis stockés. Lorsqu'un produit est demandé par un client, il est ressorti du stock (éventuellement par le procédé de picking), emballé et chargé dans un camion sortant. Dans une plateforme de cross-docking ou crossdock, cependant, les produits transitent directement des camions entrants vers les camions sortants : ils sont déchargés, triés, transférés et rechargés en moins de 24 heures, avec le minimum de stockage intermédiaire. Cette technique permet donc d'accélérer les flux et d'éliminer la majeure partie des coûts de stockage.

Cette thèse s'intéresse aux opérations logistiques qui ont lieu dans un crossdock, et aux décisions opérationnelles nécessaires pour un fonctionnement efficace de la plateforme. De nombreuses questions opérationnelles doivent être traitées au fil des opérations quotidiennes ; notre premier objectif est de dégager celles qui sont actuellement les plus critiques. Une revue de la littérature sur le sujet est donc réalisée au chapitre 1 et comparée avec les pratiques observées dans l'industrie, grâce à une grille de comparaison. Ceci nous permet d'identifier les écarts existant entre l'état de l'art et les besoins des managers de plateforme. De cette étude sont tirées les motivations de cette thèse, à savoir répondre à deux besoins identifiés comme critiques:

Comment gérer les camions en retard sans perturber le reste des opérations ?

Comment planifier le travail des employés pour traiter toutes les opérations ?

Les chapitres 2, 3 et 4 traitent la première question, tandis que le chapitre 5 répond à la seconde. Enfin, le chapitre 6 propose de combiner les deux aspects afin de traiter les deux questionnements de façon intégrée.

Le savant n'est pas l'homme qui fournit les vraies réponses ; c'est celui qui pose les vraies questions.

— Claude Levi-Strauss

Chapter 1

CONTEXT

This chapter introduces the general context and definitions of the concepts. A literature review is conducted and compared with on-field observations using a common framework. Analyzing the gaps between the state-of-the-art and the industry practice helps drawing the research questions addressed in this dissertation.

CONTEXTE

Le cross-docking, aussi appelé en français *groupage – dégroupage*, consiste en un transbordement des produits avec un minimum de stockage intermédiaire. Dans une plateforme de cross-docking (ou *crossdock*), les produits sont déchargés des camions entrants, triés, et directement rechargés pour repartir vers leur prochaine destination. Chaque produit aura passé moins de 24 heures au total dans la plateforme. En éliminant le stockage intermédiaire, cette technique permet de réduire les coûts et d'accélérer les flux, mais elle nécessite une planification rigoureuse. De nombreuses questions se posent au manager en charge des opérations: à quelle heure, à quelle porte, avec quelle ressource faut-il décharger chaque camion ? Où déplacer chaque palette, faut-il la stocker momentanément, ou faut-il au contraire prendre une palette du stock pour compléter un chargement ? Comment organiser le chargement pour que chaque camion parte à l'heure prévue ?

Ce chapitre présente le contexte général de l'étude et la définition des concepts étudiés. Grâce à une grille d'analyse commune et à une proposition de vocabulaire unifié, l'état de l'art est comparé à nos observations sur le terrain de la réalité de l'industrie. L'analyse des écarts observés permet de dégager deux axes de travail pour cette thèse, qui sont des problèmes fréquemment rencontrés dans l'industrie mais peu abordés dans la littérature : l'incertitude sur les horaires d'arrivée des camions, et la prise en compte des ressources internes.

1 CONTEXT

1.1 LOGISTICS AND CROSS-DOCKING

If you grow your own vegetables, breed your own poultry, and dress with hemp that grows nearby, transportation is not really a problem. But as soon as you start consuming products coming from further away, there arises the question: how to move items from the place where they are made to the place where they are used? This question does not only address the means of transportation, but also the organization aspects: the best time and frequency for the move, the number of items to be moved, the best path to be taken... And just like that, you are doing logistics. In order to define logistics more precisely, one can refer to the Council of Supply Chain Management Professionals:

How to move items from the place where they are made to the place where they are used?

> "*Logistics*: the process of planning, implementing, and controlling procedures for the efficient and effective transportation and storage of goods including services, and related information from the point of origin to the point of consumption for the purpose of conforming to customer requirements. This definition includes inbound, outbound, internal, and external movements".
>
> *Council of Supply Chain Management Professionals* [52]

With experience and small volumes to move, farmers until the eighteenth century managed to satisfactorily answer the question. Logistics was a complex question only for the army, who needed to equip and feed important numbers moving in potentially unknown areas. The Industrial Revolution (from about 1760/1780 to 1830/1840) and the appearance of mass production were game-changing: industries were now producing massive quantities in a single place, to be distributed to consumers located all over the country – and later, all over the world.

New means of transportation appeared, and with them new organizations. Competition, the search for new markets, and several economic crises were incentives for industries to reduce and optimize their manufacturing costs. Henry Ford addressed this question in 1908: optimizing logistic costs as well was an idea that arose very late in comparison. Ikea started designing furniture in flat packages in 1956; and it was only in 1980 that Porter [161] identified logistics as a potential competitive advantage for companies.

Ikea's idea to use flat package optimizes storage and truck loads and thus reduces logistic costs.

Cost optimization or lean techniques are now widely used in the manufacturing sector (more than 95% of big French industrial groups currently use lean management or similar concepts [151]), and the supply chain is the next sector where important savings can be achieved by implementing lean concepts. This fact was further highlighted by the development of e-commerce. Online customers are at once very demanding and very volatile: if not fully satisfied with the service one gets in a shop, it is much easier to find another online shop selling the same item, than physically going to another shop. Competition between online sellers is thus even sharper than between traditional sellers. The average quality and service level rises fast, and customers now find it normal to order shoes online and find them in their mailbox the next day – with free delivery. And if they find out the shoes are not the right size, they expect the return process to be very quick and easy.

Supply chain and logistics are essential for a company's performance.

Those new challenges, new markets, new organizations moved logistics from its old position of support function to a key position within companies. As shown by the global supply chain survey carried out by PwC in 2013 [164], companies acknowledging supply chain as a strategic asset achieve 70% higher performance than companies who do not.

> "Supply chain executives see increasing the profitability of their companies' supply chain and reducing total supply chains costs as their top priorities. In addition, more than two-thirds say it's vital to meet the requirements of customers, who are becoming more demanding about the delivery performance, flexibility and service levels they expect".
>
> *Global Supply Chain Survey 2013 by PwC* [164]

How to create lean supply chains? How to achieve a fast delivery, a good service level with a minimum cost? Cross-docking is a logistics technique that helps tackling such challenges.

1.1.1 *Cross-docking: definitions*

A plant manufacturing consumer goods tends to produce them in big batches, and thus sends full truck loads of one type of products. But a retailer hardly ever needs high volumes of a single product. A traditional way to cope with the problem is to make the products transit through a stock. The stock can be located in the manufacturer's plant, near the retailer's shop, or somewhere in between. The manufacturer can push all the production to storage while retailers pull only the needed quantity. This solution is quite flexible but has a major drawback: stock is expensive.

Cross-docking proposes an alternative solution: transferring goods directly from the truck coming from the manufacturer to several outbound trucks going to different retailers. The outbound trucks are loaded with goods coming from different manufacturers, *i. e.* different inbound trucks. On the whole, the goods stay less than 24 hours in the platform, which accelerates the flow of goods and eliminates most of the storage costs – making it a lean approach as emphasized by Cook *et al.* [51]. For a formal definition of *cross-docking*, we refer to the definition proposed by the Council of Supply Chain Management Professionals in their glossary:

> "*Cross-docking*: distribution system in which merchandise received at the warehouse or distribution center is not put away, but instead is readied for shipment to retail stores. Cross-docking requires close synchronization of all inbound and outbound shipment movements. By eliminating the put-away, storage and selection operations, it can significantly reduce distribution costs".
>
> *Council of Supply Chain Management Professionals* [52]

Cross-docking requires close synchronization of all inbound and outbound shipment movements.

We call *crossdock* the platform (also called warehouse or distribution center) where such a process takes place. Figure 1.1 shows an example of crossdock. The inbound trucks on the left-hand side contain products with different destinations (different colors). The products are unloaded, sorted, and their content is reloaded in the outbound trucks on the right-hand side heading to distinct destinations.

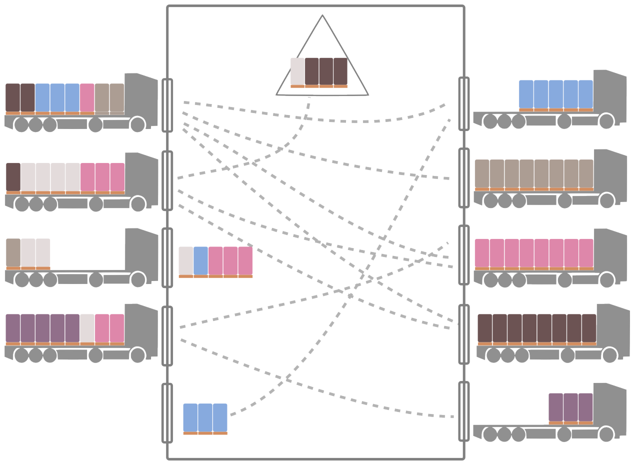


Figure 1.1: An example of crossdock

Crossdocks usually have a large number of doors, so as to accommodate several trucks at the same time. Originally, a dock is "a structure extending alongshore or out from the shore into a body of water, to which boats may be moored." [1] In the naval environment (which the word comes from) there are obviously no doors involved. It may explain why "door" and "dock" are two terms rather interchangeable in the case of a logistic platform. For instance, following the naval meaning, "docking a truck" is placing it at a given door. Strictly

1. Oxford Dictionaries. April 2010. Oxford University Press.

speaking, the door is the opening in the wall, while the dock is the area on the floor where the goods are unloaded.

1.1.2 *Cross-docking in practice*

The idea of cross-docking is about as old as postal service (or older: Ertek [69] notices that the Silk Road was a complete cross-docking operation); but Wal-Mart is often cited as the first retailer to implement it, in the late 1980's. In an analysis of Wal-Mart's success published in 1992, Stalk *et al.* evoke "a largely invisible logistics technique known as *cross-docking*" [187].

> "In this system, goods are continuously delivered to Wal-Mart's warehouses, where they are selected, repacked, and then dispatched to stores, often without ever sitting in inventory. Instead of spending valuable time in the warehouse, goods just cross from one loading door to another in 48 hours or less. Cross-docking enables Wal-Mart to achieve the economies that come with purchasing full truckloads of goods while avoiding the usual inventory and handling costs".
>
> *Stalk* et al. [187]

Running 85% of its goods through its crossdocks enabled Wal-Mart to lower its costs of sales by 2% to 3% compared to the industry average in 1992 – and to become the highest profit retailer in the world at that time (Stalk *et al.* [187]). Office Depot is another American company that achieved major gains with an early adoption of cross-docking techniques (Ross [169]).

See Ertek [68] for other examples of cross-docking implementations.

In a survey carried out in December 2010 among supply chain professionals in the United States, Saddle Creek Corporation [171] notices a significant increase of cross-docking practices between 2007 and 2010, mainly prompted by the challenging economic conditions. The greatest benefits of cross-docking according to the survey respondents are detailed in Table 1.1.

Examples of successful cross-docking implementation in Europe include Goodyear Great Britain in the 1990's: according to Kinnear [110], the new organization increased the service level (deliveries the next day increased from 87% to 96%), reduced the inventory value by 16%, released 12,500 square meters of warehousing, and reduced the operating costs by over 12%. In France, Carrefour started cross-docking fresh foods in 1994 and extended this logic to soft goods in 2009 (Rognon [166]).

Qiu *et al.* [165] show that cross-docking is particularly suitable for cold or frozen food, which requires an especially fast transportation and close to no storage. Their assertion is supported by the results of the cost analysis by Vasiljevic *et al.* [206] regarding the implementation of cross-docking to distribute food in Serbia.

Greatest benefit of cross-docking	% of respondents
Improved service level	19.4%
Reduced transportation costs	14.3%
Consolidated shipments to destination	13.1%
Get products to market more quickly	10.2%
Reduced need for warehouse space	8.5%
Improved inventory management	8%
Savings from reduced inventory carrying costs	5.7%
Increased demand for JIT service	4.5%
Shipments/consignee customization	4%
Reduced labor costs	4%
Other	8.3%

Source: Saddle Creek Corporation [171]

Table 1.1: Motivations to make the move to cross-docking

To make the move to cross-docking, a company must have a mature supply-chain organization and an efficient Information Technology (IT) system. Napolitano [153], Apte and Viswanathan [12], Gue [89] and Vogt [210] propose practical guides and a list of key success factors for cross-docking implementation. The key decisions to be made cover different time scales and different stakes: in the next section we divide them into three different levels.

1.1.3 *Three decision levels*

This dissertation focuses on the operational level, but we review quickly the other two decision levels.

The decisions to be made when planning, designing, implementing and running a crossdock cover three different levels: strategical, tactical and operational.

1.1.3.1 *Strategical decisions*

Strategical decisions are long-term decisions with a strong influence on the crossdock lifespan, and the tactical and operational decisions that follow. They are often the responsibility of the executive board. Examples of strategical decisions to be made when designing a new cross-docking system include:

THE LOCATION of the platform, geographically and within a network of suppliers, clients and other platforms. Influenced by legislation, social matters and road access, the choice depends on the position of the other actors of the network. Facility location is a widely studied problem, in which the objective is often to minimize transportation costs or duration. Facility location problems become specific to cross-docking if temporary storage is not allowed, or if the optimal flow of goods (with transshipment) within the network is used in determining the best location. For a review of articles dealing with crossdock location problems, one can refer to Van Belle *et al.* [199].

THE LAYOUT of the platform, namely the size, the shape and the number of doors. Cross-docks can have a large variety of shapes, usually described by a letter: I, L, U, T, H, E,... Doors are crucial resources in a cross-docking platform, and buildings are often built with the greatest possible number of doors, *i. e.* doors on at least two sides of the building. Sometimes the layout is simply determined by external constraints (for example the shape of the lot where the building is built, landscape integrity regulations that force all doors to be on a single side). When there are no external constraints, Bartholdi and Gue [19] study the shape that maximizes crossdock performance, *i. e.* minimizes the travel distances to transfer the goods, depending on its size (its number of doors). Although performance also depends on *e. g.* the freight flow pattern, Bartholdi and Gue's experiments suggest that an I-shape is most suited for cross-docks with fewer than about 150 doors, T-shape is best for intermediate sizes and X-shape is the most efficient for more than about 200 doors. Note that despite those results, the biggest crossdocks in France are often built by estate agents who prefer the I-shape since it can be easily split to be let to different logistic companies. Kapetanios *et al.* [109] study the performance of a crossdock depending on its number of doors, but with no precisions on the platform shape. Carlo and Bozer [40], noticing that X-shaped crossdocks can create significant congestions and safety issues, examine the optimal shape of a rectangular crossdock and the location of its "best doors".

62% of the logistic service providers in France rent their platforms [177].

1.1.3.2 *Tactical decisions*

Tactical decisions are mid-term decisions on the platform management. They are strongly influenced by the strategical decisions, and have a direct impact on the operational decisions. Examples of tactical decisions include:

THE PRODUCTS to be crossdocked. The technique is not adapted to all types of products; it is especially suited for items that are delivered frequently to a broad range of clients. Li *et al.* [124, 125, 128] propose models to help deciding which products are the most suitable for cross-docking.

THE IT SETTINGS, which are a crucial factor for a successful cross-docking system according to professionals [171]. The software used to run warehouses, called Warehouse Management System (WMS), seldom includes a cross-docking module capable of managing the case of a transfer without storage [70, 150].

Companies can also develop a custom function to monitor their cross-docking activity.

THE ROUTING OF GOODS within the cross-docking network, from the supplier to the client (through one or several cross-docking platforms). The problem is to determine the flow of goods through the network while matching supply and demand and minimizing storage (extension of the transshipment problem). Another way to look at the question is to consider vehicles instead of representing the shipments of goods as flows. It is then necessary to schedule the vehicles, and to determine pick-up and delivery times for the trucks. A detailed review of these types of problems can be found in Van Belle *et al.* [199].

THE INTERNAL LAYOUT of the platform, and especially the design of the temporary storage area(s). Ideally, the goods arriving in a crossdock are transferred directly from truck to truck: but such an organization is rarely possible for operational reasons (control/value-added operation to be done on the incoming freight, scheduling imprecision...). The goods can then be temporarily stored on the floor or in racks, after unloading or before loading (single-stage), or both (two-stages). Gue and Kang [91] use simulation to compare the different organizations and the number and size of the storage locations required in each case. In some cases (frozen food, for example), temporary storage can also be completely forbidden.

Other examples of decisions to be made at the tactical level are detailed in section 1.2.1.2.

1.1.3.3 *Operational decisions*

Operational decisions are made on a weekly or daily basis by the platform manager. Following the path of a cross-docked product,

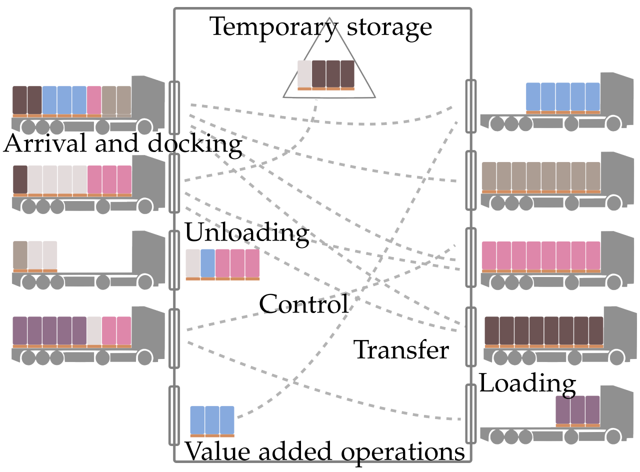


Figure 1.2: Cross-docking processes

one can list the different processes (logistic operations) occurring in a crossdock (see Figure 1.2), and the questions corresponding to different operational decisions to be made.

TRUCK ARRIVAL AND DOCKING. A truck is a tractor towing a trailer. When an inbound truck containing pallets for different destinations arrives at the platform, it awaits instructions regarding the door where it should dock. When the trailer is safely docked, the tractor can leave it without waiting for it to be loaded or unloaded, and drive back with another loaded trailer. The difference between the terms *trailer* and *truck* being slight when talking about scheduling operations, from this point on we will use the term *truck* which is more commonly used in the literature.

Operational questions:

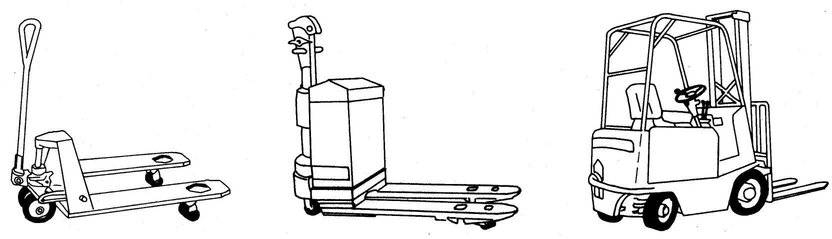
1. At what time should the truck be docked?
2. At which door should each truck be docked?

Note that such questions can be closely linked to the routing decisions mentioned in the previous section.

UNLOADING is the process of emptying the truck; its content can be put on the floor in front of the door (dock), or directly moved to the outbound truck or a temporary storage. Depending on the size of the items to be unloaded and the way they are packaged, the operation can be done manually, with a hand pallet truck, a powered pallet truck or a forklift (see Figure 1.3). Bulk goods might need to be palletized while being unloaded, and multi-reference loads must be sorted. Depending of the item type and the way the truck was loaded, the unloading might be done following a Last In – First Out (LIFO) or First In – First Out (FIFO) logic, or in any order. The truck leaves when it is empty.

Operational questions (also valid for all next operations):

3. At what time should the operation take place? How long will it last?



(a) Hand pallet truck (b) Powered pallet truck (c) Forklift truck

Source: www.fao.org/docrep/003/p3407e/P3407E10.htm – accessed 2014, April 1st.

Figure 1.3: Material handling equipment

4. Which resources (workers, equipment) should be allocated to this operation?

CONTROL. To be certain of the integrity of the data regarding the incoming freight and/or its quality, it might be necessary to control some incoming products, or all of them. The control can consist in visually checking the product, counting the unloaded pallets, scanning the items or pallets one by one...

VALUE ADDED OPERATIONS. Some other operations can be done in the cross-docking platform, to some of the freight or all of it: re-labeling, re-packing, picking from the pallets to create packages...

TEMPORARY STORAGE. As mentioned in section 1.1.3.2, there might be one or several locations for temporary storage, with different capacities, depending on the strategical decisions made earlier.

Operational questions:

5. Should a given product be temporarily stored? If so, in which storage location?
6. When should a product be taken out of storage?

TRANSFER. Again, depending on the freight type, the equipment needed for transfer from the inbound side to the outbound side can vary (see Figure 1.3). In automated platforms, conveyors are used for the transfer. A direct transfer to the outbound truck is possible only if the corresponding truck is docked. The time needed for the transfer can depend on the product weight, the distance to be crossed (often called *travel distance*), the platform congestion.

Operational questions:

7. Where to put each product: in the outbound truck, in the temporary storage?

LOADING. When an outbound truck is full, it is closed, sealed, and can leave the dock. Depending on the client's requirements, loading might have to be done in a given order.

Operational questions:

8. At what time should each truck leave?

These eight different questions, although tackled together by logistic managers, are considered separately in the literature. The next section compares industrial practices with the existing cross-docking operations literature.

1.2 STATE-OF-THE-ART AND INDUSTRY PRACTICE

The objective of this section is not only to give a literature review, but also to provide an industrial benchmark in order to identify the gaps between the current state-of-the-art and the observed industry practice.

In order to compare the problems studied in the literature with the real-life situations occurring in industry, a common reference grid is necessary. Section 1.2.1 details the comparison framework used to classify both the articles found in the literature in section 1.2.2, and our on-field observations in section 1.2.3. Finally, section 1.2.4 discusses the common points and gaps observed between current research and industry practice.

1.2.1 *Comparison framework*

This section describes the different elements used to characterize a cross-docking platform and its performance indicators.

Our focus is on the platform internal operations.

Our focus is on the decisions to be made on a daily or weekly basis about the internal operations of the cross-docking platform. However, decisions made earlier on a mid-term or long-term time scale have a key impact on the operations. Therefore, we need to take strategical (long term) and tactical (mid-term) levels into account as well: in this comparison framework, they are comparison elements to identify the type of crossdock under consideration. They consist in constraints imposed by either the physical features of the platform, or tactical decisions that will not be questioned at this point, or external stakeholders.

Most of the elements listed as platform settings are introduced by Boysen and Fliedner [31] in their classification of truck scheduling problems, and re-used by Van Belle *et al.* [199] to categorize the articles of their review. The words in italics correspond to the possible values for each criterion.

1.2.1.1 *Platform settings: strategical level*

We consider situations where the physical characteristics of the platform are fixed. Following Van Belle *et al.* [199], the following characteristics are used to characterize the platform:

SHAPE, described by a letter (I, L, U, T, H, E, X...) as explained in section 1.1.3.1.

NUMBER OF DOORS and how these doors are placed along the platform (on one side only or more).

INTERNAL TRANSPORTATION. The goods inside the platform can be moved either *manually* (*e.g.* by workers using pallet trucks or forklifts) or with an *automated* system such as a network of conveyor belts. A *combination* of these two transportation modes is also possible.

1.2.1.2 *Platform settings: tactical level*

The tactical decisions or policies are also important to classify a cross-docking platform. The different characteristics are as follows:

SERVICE MODE. As defined by Boysen and Fliedner [31], the door mode is *exclusive* if each door is dedicated to inbound trucks, or outbound trucks exclusively. The door mode is *mixed* if a truck can be docked at any door. The *combination* mode occurs when some doors follow an exclusive mode of service while the others are used in mixed mode. Note that in an exclusive mode of service, it is also possible to allocate each destination to one specific outbound door, such as each outbound door serves a fixed set of destinations. We call this mode a *destination exclusive* mode of service. Oh *et al.* [158] study the tactical problem of assigning destinations to doors in a destination exclusive mode of service.

PREEMPTION. The preemption is allowed if the loading or unloading of a truck can be interrupted. The truck is then moved to a parking area to let another truck be processed at the door. The interrupted operation must be completed later, possibly at another dock.

TEMPORARY STORAGE CAPACITY. If the corresponding truck is not available when a product is unloaded, it is put in a storage area for a short period. This area may have a *limited* capacity, if the space available is scarce; otherwise the capacity can be considered as *unlimited* (∞). It is also possible, for instance in the case of frozen goods, that the products are not allowed to stay temporarily in the platform, in which case we consider the storage capacity as *zero*.

INTERNAL RESOURCE CAPACITY. The capacity of the conveyor belts network in an automated transportation mode, or the maximum number of workers available if the transportation is done manually, can be either considered *limited* or *unlimited* (∞).

1.2.1.3 *Platform settings: operational level*

Some operational characteristics of the cross-docking systems are not driven by the manager of the platform but imposed by the external stakeholders, namely the client or the transportation providers. Although those characteristics play a crucial role in the daily operations, they are not a decision variable the manager can easily adjust – even if he may be able to negotiate if needed.

PRODUCT INTERCHANGEABILITY. Products are interchangeable if one can be used instead of another for a given type of products. When products are interchangeable, two cases may occur: if each outbound truck has a list of product types that should be loaded in, it is a *post-distribution*. The second case is when each product unloaded in a cross dock is dedicated to a specific *destination*. In both cases, defining the exact product-truck allocation remains an operational decision. When the products are not interchangeable, each product is dedicated to a specific outbound truck, thus the dispatching information is known from the inbound side: it is a *pre-distribution*. See Yan and Tang [217] and Tang and Yan [191] for a comparative study of pre- and post-distributions.

ARRIVAL TIME. If any truck can be unloaded at any time, then it is not restrictive to assume that all trucks are available from the beginning of the planning horizon (time *zero*), ready to be processed at any time. On the contrary, if truck arrivals are subject to external constraints (for instance if the trucks have other appointments prior to their arrival in the platform), then the arrival times are defined *per truck*. Note that this applies for both the inbound and outbound trucks.

DEPARTURE TIME. There may be *no* restrictions on the departure time of the trucks. But if a truck has another transportation task scheduled after its departure from the crossdock, there will be a deadline for its departure. This deadline can be defined for the *inbound trucks*, *outbound trucks* or *both*. It is also possible that each *product* has a specific time before which it should leave the platform. In this case, the deadline is expressed on a product level instead of a truck level.

TRUCK FILLING. All the inbound trucks should be fully unloaded before leaving the platform; but when the outbound trucks have deadlines, they may leave the platform at the scheduled deadline without being fully loaded. In this case we say that a *Less than Truckload (LTL)* departure is allowed. Otherwise, the trucks leave only when they are *full*.

1.2.1.4 *Performance measures*

The performance of the cross-docking operations can be measured by many different indicators. Here we list different possible performance measures (which might be called elements of the objective function for an optimization problem). We include all the objectives mentioned by Boysen and Fliedner [31] (marked with * in the list) who focus only on the truck scheduling problem. Their list is completed with other objectives related to the workers, specific operations inside the platform, or truck filling rate.

INVENTORY LEVEL. Since one of the cross-docking objectives is to reduce the inventory, it is logical to follow some indicators on the inventory level, such as the *total** or the *maximum number of products stocked** in the planning horizon.

WORKING HOURS. The manpower is very often the first cost center of a logistic platform where the operations are done manually. Therefore the *total number of working hours* used to complete the operations on the planning horizon is an important indicator.

TRAVEL DISTANCE. The previous indicator can be closely linked to the *total distance traveled* by all the products inside the platform: a longer distance to be crossed requires a longer time for a worker to complete his task.

CONGESTION. Minimizing the travel distance can lead one to group all the loading and unloading tasks in the same area, which is likely to generate congestion and on the overall, slow down the process. There are no straightforward ways to measure the congestion, but the *percentage of total space used*, or the *total number of times two products cross* each other, are possible indicators.

TOTAL PRODUCT STAY TIME. If the main objective is to maximize the turnover of goods, a meaningful indicator to monitor is the *total time spent inside the platform** ("completion time" for Boysen and Fliedner) for all the products.

TOTAL LOADING OR UNLOADING TIME. In order to accelerate the turnover of goods and free the doors as soon as possible, minimizing the *total time spent at the outbound docks* by the outbound trucks is a possible objective. Similarly, if the inbound door utilization rate is high, the *total time spent at the inbound docks* by the inbound trucks is a meaningful indicator to monitor.

TRUCK PROCESSING TIME DEVIATION. When arrival time or deadlines are defined, it is important to ensure that they are respected, with an indicator on the *earliness* or *tardiness** of the inbound or outbound trucks. Note that we are not talking here about the punctuality of the transportation provider. Although very important, it is not directly influenced by the operations management. The indicator discussed keeps track of situations when the trucks are forced to arrive earlier or leave later than planned, because it is not possible to start their unloading or finish their loading on time.

DOOR UTILIZATION. An indicator closely linked to the total loading or unloading time is the inbound or outbound *door utilization rate*.

TRUCK FILLING RATE. If a less-than-truckload departure is allowed, it is reasonable to keep track of the *filling rate* of the truck, in order to ensure that the cost savings by cross docking are compensated for by increased transportation costs due to half-full trucks.

PRODUCTS NOT LOADED. Another indicator that can be monitored when less-than-truckload departures are allowed is the *number* *of missed orders*, *i.e.* the number of products that could not be loaded, or the corresponding *lost profit*.

SCHEDULE LENGTH. If an important goal is to finish the operations as early as possible, the total schedule length or *makespan** can be monitored. It is the point in time at which the last operation (possibly the last truck load) is completed.

1.2.2 *Literature analysis*

Since the scope of our analysis is the operational level of cross-docking problems, we put aside of this literature review all the problems at strategic or tactical levels mentioned in sections 1.1.3.1 and 1.1.3.2, such as network design or truck routing. We focus on the operations taking place at the platform. The literature review by Van Belle *et al.* [199] cites 42 papers that enter this scope, the more recent being published in 2011. Because the comparison framework proposed in section 1.2.1 is largely inspired by the characteristics proposed by Van Belle *et al.*, this section includes the articles studied in their review and complements it with additional articles.

1.2.2.1 *Methodology and problem classification*

We reviewed only articles written in English. Besides *crossdock* and *cross-docking* [2], we searched the key words *transshipment*, *dispatch*, *Less than Truckload (LTL) terminal*, *breakbulk terminal*, *yard management*. The articles found were filtered to keep only those dealing with the operational level. On the whole, the review includes 120 articles from different sources detailed in Table 1.2, which means that we add 78 papers to the 42 papers cited by Van Belle *et al.* [199].

Source	Number of papers
Journal	64
Conference	31
Book chapter	8
Master thesis	7
Technical report	6
PhD thesis	4

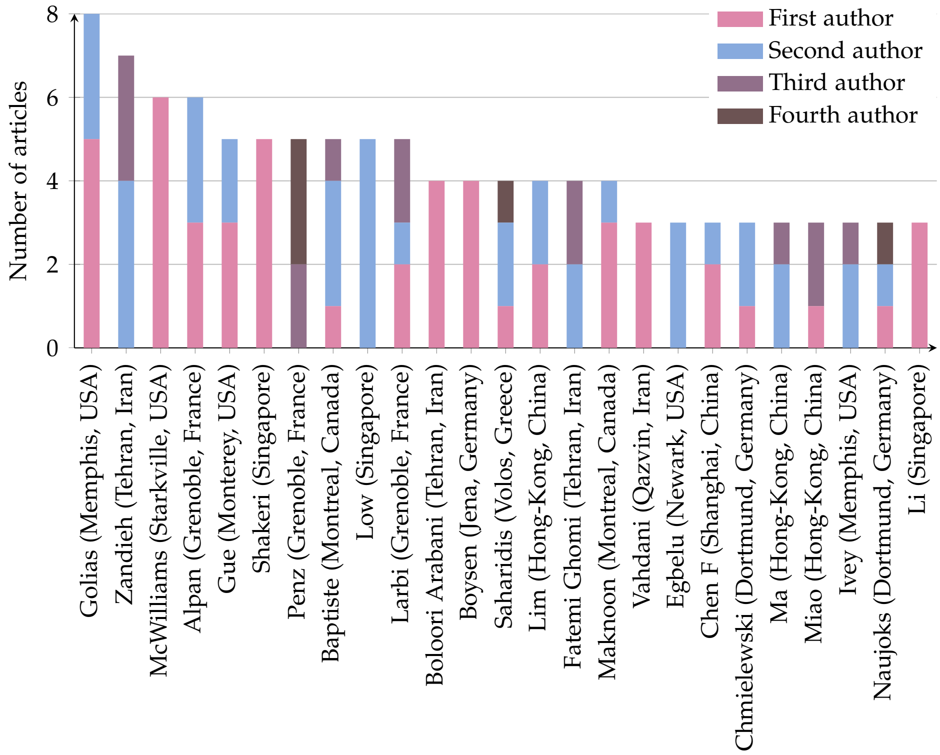
Table 1.2: Source type for the different articles reviewed

The titles of the journals in which the 64 journal articles were published are summarized in Table 1.3. Figure 1.4 gives an overview of the authors who contributed most significantly to the field, with their geographical location.

2. and their variations *cross dock*, *cross-dock*, *crossdocking*, *cross docking*.

Journal title	Number of articles
Computers & Industrial Engineering	15
Computers & Operations Research	5
The International Journal of Advanced Manufacturing Technology	5
European Journal of Operational Research	4
Transportation Science	3
Journal of Intelligent Manufacturing	2
Journal of the Operational Research Society	2
Transportation Research Part E: Logistics and Transportation Review	2
Applied Soft Computing	2
International Journal of Information Systems and Supply Chain Management	2
International Journal of Production Economics	2
International Journal of Production Research	2
OR Spectrum	2
Transportation Research Record	1
Journal of Engineering Manufacture	1
Operations Research	1
Expert Systems with Applications	1
Transportation Journal	1
Annals of Operations Research	1
International Journal of Logistics Systems and Management	1
Journal of American Science	1
Journal of Industrial and Systems Engineering	1
Interfaces	1
Computers & Chemical Engineering	1
Journal of Service Science and Management	1
International Journal of Industrial Engineering Computations	1
International Journal of Logistics Research and Applications	1
IIE Transactions	1
Omega	1

Table 1.3: Source of the 64 journal articles reviewed



Only the authors of more than 3 articles in total are presented – the city and country are taken from their most recent reviewed article.

Figure 1.4: Main contributing authors

Cross-docking being a comparatively recent research field, it seems that no standard names have been set for different optimization problems. In the titles of all the papers cited in the present article, we could find different terms qualifying the problems at hand. The number of occurrences of those different terms are displayed in Table 1.4. Note that the total number of papers in Table 1.4 does not equal the total number of papers reviewed, because some of them do not include any of these expressions in their titles.

(a)	(b)	(c)
Door assignment (9) Truck dock assignment (3) Dock door assignment (3) Dock assignment (2) Truck-to-door assignment (1) Trailer to door assignment (1)	Truck scheduling (16) Trailer scheduling (2) Truck sequencing (1)	Crossdock scheduling (10) Crossdock operations scheduling (4)

Table 1.4: Occurrence of different problem names in the titles of the papers reviewed

Most papers in column (a) of Table 1.4 consider a set of doors (inbound, outbound or both) and a set of trucks, the number of trucks being less than or equal to the number of doors. The problem is therefore restricted to a single moment in time, and the question is to choose at which door each truck present at that time should be placed. We will therefore refer to this problem as a *truck-to-door assignment problem*. Even if this term is used only once (by Shakeri *et al.* [180]) in the set of papers studied, we believe that it is the most descriptive name for this problem.

Papers listed in column (b) of Table 1.4 add a time dimension to the previous problem. If there are more trucks than doors (*i.e.* if we do not fall into the truck-to-door assignment problem), then it is necessary to assign more than one truck to each door; therefore, at each door the trucks should be sequenced in time. We call this a *truck-to-door sequencing problem*. If the model determines exact arrival/departure hours instead of the order in which the trucks arrive, we call it a *truck-to-door scheduling problem*. We distinguish these from another problem which aims at determining at what time the trucks are docked, without specifying the exact dock. We call the latter a *truck sequencing problem* or *truck scheduling problem*. Note that the truck sequencing/scheduling problem can be solved in sequence with the truck-to-door assignment problem in order to decide firstly at what time, and secondly at which door the trucks are docked.

The terms "crossdock scheduling" and "crossdock operations scheduling", found in the titles of 13 articles as shown in column (c) of Table 1.4 are rather general. Looking more closely, we can see that all the papers mentioning "crossdock scheduling" in their titles are actually dealing with truck scheduling or sequencing problems. The papers about "operations scheduling" deal with (outbound) truck sequencing, truck scheduling or truck-to-door scheduling problems.

Table 1.5 summarizes the definition of the different classes of problems which we propose. The N/A symbol indicates criteria that are not applicable to the problem.

	Which door?	What time?	In which order?
Truck-to-door assignment	✓	N/A	N/A
Truck-to-door sequencing	✓	N/A	✓
Truck-to-door scheduling	✓	✓	N/A
Truck sequencing	N/A	N/A	✓
Truck scheduling	N/A	✓	N/A

Table 1.5: Summary of the different operational cross-docking problems

Historically, as shown in Figure 1.5, the first problems studied by the research community are the truck-to-door assignment and the truck sequencing problems, which are somewhat simpler. The first paper considering a time horizon on this problem was published in 2002 (Yu [219]), and the first paper dealing with truck scheduling appeared in 2009 (Chen *et al.* [44]). The interest of the community is now equally spread between the four more complex problems: truck sequencing/scheduling and truck-to-door sequencing/scheduling.

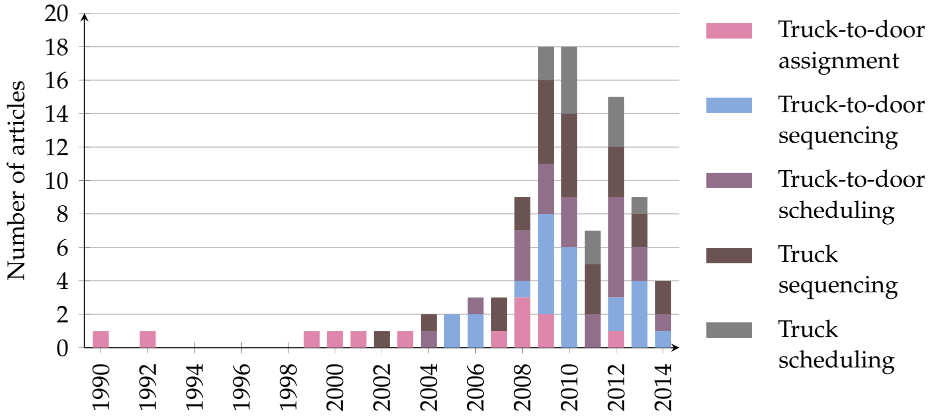


Figure 1.5: Number of articles per year

Table 1.6 gives an overview of the solution methods used in the different articles – note that one article can use more than one solution method, thus some references appear several times in the table.

The next sections present the reviewed papers in each of the problem categories proposed. The type of platform and the type of problem studied in each paper are qualified according to the framework detailed in section 1.2.1. In all the tables displaying the results, the symbol v following a citation indicates that the article was already present in the review by Van Belle *et al.* [199]. Similarly to the notation they adopted, the symbol * in the body of the table indicates that the model studied is applicable to any non-zero value of the criterion, "n/a" means that the criterion is not applicable to the problem at hand, while "ns" means the information was not specified in the article studied.

As noted already by Van Belle *et al.*, some papers cited in one of the categories proposed also make use of simulation to test their solution approaches. Indeed, simulation is often used in order to gain insights on complex operational problems within a crossdock. More details on crossdock-related papers using simulation will be given in chapter 3.

1.2.2.2 *Truck-to-door assignment*

The truck-to-door assignment, sometimes also called *yard management*, consists in allocating trucks to doors at a given point in time. The papers dealing with this problem are listed in Table 1.7.

	Solution method	Articles using the method
Exact methods	Polynomial algorithm	[44], [115], [172], [208], [209]
	Mathematical programming	[1], [2], [3], [11], [22], [33], [29], [34], [42], [43], [46], [47], [63], [71], [105], [132], [131], [138], [137], [142], [144], [149], [155], [157], [159], [168], [174], [180], [179], [178], [182], [186], [189], [193], [194],[200], [218], [220], [221], [222], [223]
	Dynamic programming	[8], [10], [9], [16], [33], [28], [29], [30], [116], [136], [138], [172], [173]
Heuristics	IP-based heuristics	[4], [46], [189], [214]
	Scheduling heuristics	[43], [44], [181], [186]
	Other dedicated heuristics	[1], [2], [9], [16], [21], [33], [28], [29], [30], [35], [49], [50], [54], [85], [88], [99], [102], [111], [113], [116], [115], [123], [127], [141], [143], [157], [159], [175], [178], [182], [192], [193], [194], [212], [216], [218], [220], [221], [222]
Metaheuristics	Evolutionary algorithm	[20], [24], [25], [26], [27], [46], [81], [107], [129], [130], [136], [155]
	Genetic algorithm	[23], [24], [25], [26], [47], [57], [71], [73], [83], [96], [107], [111], [113], [112], [120], [122], [132], [131], [145], [146], [142], [149], [183], [197], [195], [216], [218]
	Tabu search	[11], [16], [25], [130], [132], [197], [200]
	Simulated annealing	[18], [28], [30], [34], [130], [134], [140], [185], [197]
	Particule swarm optimization	[24], [25], [26]
	Memetic algorithm	[82], [84], [106]
	Local search	[35], [96], [185], [197], [218]
	Other metaheuristic	[11], [73], [93], [92], [130], [134], [197], [195], [196]

Table 1.6: Solution methods for the reviewed articles

			Strategical level				Tactical level	Perf. measures	
		On which doors	Shape	Nb inbound doors	Nb outbound doors	Internal transport	Service mode	Distance traveled	Congestion
Bartholdi and Gue	[18][v]	both	I	*	*	*	Destination excl.	✓	✓
Bermúdez and Cole	[23][v]		*	*	*	Manually	Exclusive	w	
Bozer and Carlo	[34][v]	both	*	*	*	Manually	Mixed	✓	✓
Brown	[35][v]	both	*	*	*	Manually	Exclusive	total	
Cohen and Keren	[49]	both	I	*	*	Manually	Exclusive	w	
Cohen and Keren	[50][v]	both	I	*	*	Manually	Exclusive	✓	
Gue	[88][v]	inb.	I	*	*	Manually	Exclusive	w	
Guignard *et al.*	[93]	both	*	*	*	Manually	Destination excl.	✓	
Jarrah *et al.*	[105]	outb.	E	*	*	Automated	Exclusive		
Ko *et al.*	[111]	outb.				Combination	Destination excl.		
Ley and Elfayoumy	[120]	both	I	*	*	ns	Exclusive	✓	
Peck	[159][v]		I	*	*	Manually	Exclusive		
Tsui and Chang	[193][v]		I	*	*	Manually	Exclusive	✓	
Tsui and Chang	[194][v]		I	*	*	Manually	Exclusive	✓	
Yu *et al.*	[218][v]	inb.	*	*	*	Manually	Destination excl.	✓	
Zhu *et al.*	[223]	both	*	*	*	Manually	Exclusive	✓	

Table 1.7: Truck-to-door assignment

As shown in Figure 1.5, this problem was mainly studied before 2010, which explains why most papers are listed in the review by Van Belle *et al.* [199].

The case treated by Guignard *et al.* [93] is another version of the problem which consists in assigning, for a given horizon, origins and destinations to the different doors. This version considers a time horizon and not a single moment in time, but aggregates all the data of this horizon so that the time dimension is not explicitly considered.

Similarly, Ko *et al.* [111] assign destination to outbound doors in a parcel sorting platform. But the objective function used is very different from the others (thus it does not appear on the table): they work with the double objective of minimizing the number of working teams for loading the outbound freight, and balancing the workload between the different teams.

Jarrah *et al.* [105] also study a parcel sorting platform in which destinations should be assigned to doors; however, from time to time this destination-to-door assignment should be modified. Their first objective is to minimize the occurrences of such changes, the second is to minimize the number of workers assigned to the loading operation, and the third one is to evenly distribute the parcel at the different loaders. Once again, the elements of the objective function being very original compared to the rest of the truck-to-door assignment literature, they are not displayed in Table 1.7.

1.2.2.3 *Truck-to-door sequencing*

The truck-to-door sequencing problem assigns trucks to doors, but on a time horizon rather than on a given point in time. Since there are more trucks than doors (otherwise the problem is reduced to the truck-to-door assignment problem), a given door is assigned to several trucks, for which a processing order has to be determined. Table 1.8 shows the articles dealing with this problem.

This hard problem cannot be reduced to a 1-inbound, 1-outbound door case without making the door allocation trivial (it would thus become a truck sequencing problem). In order to simplify the problem, about half of the authors consider only the inbound side, the outbound side being either constraint-free or with fixed departures.

The distance traveled is an objective inherited from the truck-to-door allocating problem, and it does not take into account the trucks point of view which is incorporated with objectives such as the truck time deviation, loading time or unloading time. If the speed is considered constant, which is a common assumption, then minimizing the distance traveled amounts to minimizing the total travel time. As an aggregating measure for the distance traveled and other time related indicators, the makespan is thus also a popular performance measure.

1.2.2.4 *Truck-to-door scheduling*

The truck-to-door scheduling problem also consists in allocating trucks to doors on a time horizon. The difference with the truck-to-door sequencing problem is that the former does not model the time explicitly since it considers only the order in which the different trucks are processed at the dock. The papers related to the truck-to-door scheduling problem, listed in Table 1.9, model the time dimension in an explicit way.

Again, the distance traveled and the makespan are popular performance measures. Making the time dimension explicit also allows one to follow the inventory level as a performance measure.

Chmielewski *et al.* and Naujoks and Chmielewski [46, 155] deal with a variation of the problem that consists in allocating the door role and destinations in a destination-exclusive crossdock. They look for an optimal allocation of resources (workers, scanners, forklifts, *etc.*), while minimizing firstly the total travel distance, and secondly the trucks waiting time.

Two very recent papers appearing in Table 1.9 (Agustina *et al.* [4] and Dondo and Cerdá [63]) actually deal with a more complex problem, which puts together truck routing (how to move products from the suppliers to the platform, and from the platform to the clients, with a limited amount of trucks) and truck scheduling.

			Strategical level				Tactical level				Operational level				Performance measures								
		On which doors	Shape	Nb inbound doors	Nb outbound doors	Internal transport	Service mode	Pre-emption	Storage capacity	Resources capacity	Arrival time	Departure time	Truck filling	Interchangeability	Distance traveled	Congestion	Total product stay time	Truck time deviation	Loading time	Unloading time	Door utilization	Products not loaded	Makespan
Boysen and Fliedner	[31][v]	both	*	*	*	ns	Excl.	No	0	∞	zero	Outb.	LTL	Dest.								✓	
Boysen *et al.*	[29]	inb.	*	*	*	Manually	Excl.	No	0	∞	zero	Outb.	LTL	Pre-D								✓	
Boysen *et al.*	[30]	inb.	*	*	*	Manually	Excl.	No	0	∞	zero	Outb.	LTL	Pre-D								✓	
Dondo and Cerdá	[63]	both	*	*	*	Manually	Excl.	No	∞	∞	zero	No	full	Pre-D									✓
Golias *et al.*	[81]	both	*	*	*	ns	n/a	No	∞	∞	/truck	Both	full	Pre-D				✓		✓			
Golias *et al.*	[82]	both	*	*	*	Manually	Excl.	No	∞	∞	/truck	No	n/a	Pre-D	✓		✓						
Golias *et al.*	[83]	inb.	I	*	ns	Manually	Excl.	No	ns	n/a	/truck	Inb.	n/a	Dest.						✓			
Golias *et al.*	[84]	both	*	*	*	Manually	Excl.	No	0	∞	/truck	Both	full	Pre-D				✓		✓	✓		
Ji	[106]	both	*	*	*	Manually	Excl.	No	∞	∞	/truck	No	n/a	Pre-D	✓		✓						
Konur and Golias	[113]	inb.	ns	*	ns	Manually	Excl.	No	n/a	∞	/truck	No	n/a	n/a				✓	✓				
Konur and Golias	[112]	inb.	*	*	*	Manually	Excl.	No	n/a	∞	/truck	No	n/a	n/a				✓	✓				
Liao *et al.*	[130]	inb.	*	*	big	Manually	Excl.	No	∞	∞	zero	Outb.	LTL	Dest.								✓	
Lim *et al.*	[132][v]	both	*	*	*	Manually	Mixed	No	n/a	lim	/truck	Both	n/a	Pre-D	✓								
Lim *et al.*	[131][v]	both	*	*	*	Manually	Mixed	No	n/a	lim	/truck	Both	n/a	Pre-D	✓								
Madani-Isfahani *et al.*	[134]	both	*	*	*	ns	Excl.	?	∞	∞[3]	zero	No	full	Dest.									✓
McWilliams *et al.*	[145][v]	inb.	*	*	*	Automated	Excl.	No	0	lim	zero	No	full	Dest.									✓
McWilliams	[140]	inb.	*	*	*	Automated	Excl.	No	0	lim	zero	No	full	Dest.									✓
McWilliams *et al.*	[146][v]	inb.	*	*	*	Automated	Excl.	No	0	lim	zero	No	full	Dest.									✓
McWilliams	[142][v]	inb.	*	*	*	Automated	Excl.	No	0	lim	zero	No	full	Dest.									✓
McWilliams	[141][v]	inb.	*	*	*	Automated	Excl.	No	0	lim	zero	No	full	Dest.									✓
McWilliams	[143]	inb.	*	*	*	Automated	Excl.	No	0	lim	zero	No	full	Dest.		✓							✓
McWilliams and McBride	[144]	inb.	*	*	*	Automated	Excl.	No	0	lim	zero	No	full	Dest.									✓
Miao *et al.*[v]	[149][v]	both	*	*	*	Manually	Mixed	No	n/a	lim	/truck	Both	n/a	Pre-D	✓								
Nourmohammasi Sharabiani	[157]	both	*	*	*	Manually	Excl.	No	∞	∞	zero	No	full	Post-D									✓
Rosales *et al.*	[168][v]	inb.	*	*	*	Manually	Excl.	No	n/a	lim	zero	No	n/a	Pre-D	✓								
Saharidis *et al.*	[174]	inb.	*	*	*	Manually	Excl.	No	n/a	∞	/truck	No	n/a	n/a				✓		✓			
Zhang	[221]	both	I	*	*	Manually	E+M	No	∞	lim	/truck	No	full	Pre-D	✓				✓	✓			
Zhang *et al.*	[222]	both	I	*	*	Manually	E+M	No	∞	lim	/truck	No	full	Pre-D	✓				✓	✓			

Table 1.8: Truck-to-door sequencing

[3] The status of this article is unclear regarding capacity constraints. It claims in the abstract and in the introduction to be making use of capacity constraints, but in the assumptions says that "capacity is unlimited"; then in the list of data, C is introduced as the capacity of the crossdock but never used in the mathematical model.

			Strategical level				Tactical level				Operational level				Performance measures								
		On which doors	Shape	Nb inbound doors	Nb outbound doors	Internal transport	Service mode	Pre-emption	Storage capacity	Resources capacity	Arrival time	Departure time	Truck filling	Interchangeability	Inventory level	Distance traveled	Congestion	Total product stay time	Truck time deviation	Loading time	Unloading time	Door utilization	Makespan
Acar	[1]v	inb.	ns	*	nc	Manually	Excl.	No	n/a	∞	/truck	No	n/a	n/a								✓	
Acar *et al.*	[2]	inb.	ns	*	nc	Manually	Excl.	No	n/a	∞	/truck	No	n/a	n/a								✓	
Agustina *et al.*	[3]	both	*	*	*	Manually	Excl.	No	∞	∞	/truck	Both	full	Pre-D	✓				✓				
Agustina *et al.*	[4]	both	*	*	*	Manually	Excl.	No	∞	∞	/truck	Both	full	Pre-D	✓				✓				
Bartz-Beielstein *et al.*	[20]	both	ns		25	Manually	Mixed	No	0	∞	/truck	No	full	Dest.		✓			✓				
Chmielewski *et al.*	[46]v	both	*	*	*	Manually	Dest.	No	lim	lim	/truck	Both	n/a	Dest.		✓				✓	✓		
Guignard and Hahn	[92]	both	I	25	16	Manually	Excl.	No	∞	∞	/truck	No	n/a	Dest.		✓	✓						✓
Guo *et al.*	[96]	both	*	*	*	Manually	Excl.	No	∞	∞	/truck	Outb.	full	Dest.		✓			✓				
Hermel	[102]	both	*	*	*	Manually	Mixed	No	0	lim	zero	No	full	Pre-D		✓							✓
Li *et al.*	[126]	both		*	*	Manually	Mixed	No	∞	∞	zero	No	full	Dest.									✓
Naujoks and Chmielewski	[155]	both	*	*	*	Manually	Dest.	No	lim	lim	/truck	Both	n/a	Dest.		✓				✓	✓		
Saharidis *et al.*	[174]	inb.	*	*	*	Manually	Excl.	No	n/a	∞	/truck	No	n/a	n/a					✓		✓		
Shakeri *et al.*	[180]v	both	*	*	*	Manually	Mixed	No	∞	∞	zero	No	n/a	Pre-D									✓
Shakeri *et al.*	[181]	both	*	*	*	Manually	Mixed	No	∞	∞	zero	No	n/a	Pre-D									✓
Shakeri and Low	[179]	both	*	*	*	Manually	Mixed	No	lim	lim	zero	No	full	Pre-D									✓
Shakeri	[178]	both	*	*	*	Manually	Mixed	No	∞	lim	zero	No	full	Pre-D									✓
Shakeri *et al.*	[182]	both	I	*	*	Manually	Mixed	No	0	lim	zero	No	full	Dest.									✓
Tesch *et al.*	[192]		L	14	80	Manually	Excl.	No	lim	lim	/truck	No	full	Pre-D				✓	✓				
Van Belle *et al.*	[200]	both	*	*	*	Manually	Excl.	No	∞	∞	/truck	Both	n/a	Pre-D		✓			✓				
Wang and Regan	[212]v	inb.	*	*	*	Manually	Excl.	No	0	∞	/truck	No	full	Dest.		✓					✓	✓	
Werners and Wülfing	[214]v	outb.	U	*	*	Combination	Excl.	No	0	lim	ns	Outb.	full	Dest.		✓							

Table 1.9: Truck-to-door scheduling

1.2.2.5 *Truck sequencing*

Contrary to truck-to-door problems, truck sequencing problems do not take the space dimension into account. A truck is not allocated to a specific door, but to any door as long as the total number of doors is respected. The notion of distance between two doors inside the platform, which was central for the truck-to-door assignment problem, is not considered here. The truck sequencing problem only looks at the order in which the trucks arrive at the doors. The related papers are listed in Table 1.10.

Not taking the space dimension into account allows one to simplify the problem into a 1-inbound, 1-outbound door situation. This is not a realistic assumption but it can help to understand better this difficult problem: 17 out of the 26 articles reviewed in this category make such an assumption.

The makespan is the unique performance measure considered in half the articles. The inventory level is also an important aspect for the truck sequencing problem, which is consistent with the cross-docking concept for which reducing inventory is one of the main objectives.

1.2.2.6 *Truck scheduling*

Truck scheduling takes the time dimension explicitly into account, rather than implicitly through the truck processing order. Articles dealing with such problems are displayed in Table 1.11.

Again, 5 articles out of 12 assume a crossdock with one inbound and one outbound door. Modeling the time explicitly enables one to take into account the truck time deviation, which is thus an important performance measure for the problem, besides the makespan.

1.2.2.7 *Internal operations*

Maknoon *et al.* [138] suppose the truck schedule known in a 1-inbound, 1-outbound door platform, and optimize the moving pattern of products inside the platform. The goal is to determine whether the unloaded items should go directly to the outbound truck or rather to the storage location for a later truck, in order to maximize the number of direct transfers (minimizing the number of products put in storage).

The location of temporary storage area (or staging area) is also a problem addressed in the literature. See Van Belle *et al.* [199] for a classification and review of the related works.

1.2.3 *On-field observations*

This section gives an account of visits made in eight different logistic platforms in France (1 near Paris, 5 near Lyon, 1 near Grenoble and

		Strategical level					Tactical level				Operational level				Performance measures				
		On which doors	Shape	Nb inbound doors	Nb outbound doors	Internal transport	Service mode	Pre-emption	Storage capacity	Resources capacity	Arrival time	Departure time	Truck filling	Interchangeability	Inventory level	Number of touch	Truck time deviation	Makespan	Preemption costs
Alpan *et al.*	[8]	outb.	n/a	*	*	ns	Excl.	Yes	∞	∞	zero	No	full	Dest.	✓				✓
Alpan *et al.*	[10][v]	outb.	n/a	*	*	ns	Excl.	Yes	∞	∞	zero	No	full	Dest.	✓				✓
Alpan *et al.*	[9]	outb.	n/a	*	*	Manually	Excl.	Yes	∞	∞	/truck	No	full	Dest.	✓				✓
Baptiste and Maknoon	[16]	both	n/a	1	1	Manually	Excl.	No	∞	∞	zero	No	full	Dest.		✓			
Chen and Lee	[42][v]	both	n/a	1	1	ns	Excl.	No	∞		zero	No		Pre-D				✓	
Chen and Song	[43][v]	both	*	*	*	ns	Excl.	No	∞		zero	No		Pre-D				✓	
Davoudpour *et al.*	[57]	both	n/a	1	1	Manually	Excl.	No	0	∞	/truck	Both	full	Post-D				✓	
Fazel Zarandi *et al.*	[71]	both	*	1	1	Automated	Excl.	Yes	∞	∞	/truck	Outb.	full	Dest.			✓		✓
Forouharfard and Zandieh	[73][v]	both	n/a	1	1	ns	Excl.	No	∞	∞	zero	No	full	Post-D	✓				
Ghobadian *et al.*	[79]	both	n/a	1	1	ns	Excl.	No	∞	∞	zero	No	full	Dest.				✓	
Joo and Kim	[107]	both	I	*	*	Manually	Excl.		0	∞	zero	No	full	Post-D				✓	
Larbi *et al.*	[116]	outb.	*	*	*	Manually	Excl.	Yes	∞	∞	zero	No	n/a	Dest.	✓				✓
Larbi *et al.*	[115][v]	outb.	n/a	1	1	ns	Excl.	Yes	∞	∞	zero	No		Dest.	✓				✓
Liao *et al.*	[129]	both	n/a	1	1	*	Excl.	No	∞	∞	zero	No	full	Post-D				✓	
Maknoon and Baptiste	[136]	both	*	1	1	Manually	Excl.	No	∞	∞	zero	No	full	Dest.		✓			
Maknoon and Baptiste	[137]	both	*	*	*	Manually	Excl.	No	∞	∞	/truck	Both	full	Dest.	✓				
Maknoon *et al.*	[135]	outb.	n/a	*	*	Manually	Excl.	No	∞	∞	/truck	No	full	Pre-D	✓				
Sadykov	[172]	both	n/a	1	1	*	Excl.	No	lim	∞	zero	No	full	Post-D	✓				
Sadykov	[173]	both	n/a	1	1	*	Excl.	No	lim	∞	zero	No	full	Post-D	✓				
Shiguemoto *et al.*	[183]	both	*	1	1	Automated	Excl.	No	∞	∞	zero	No	full	Dest.				✓	
Soltani and Sadjadi	[185][v]	both	n/a	1	1	Automated	Excl.	Yes	0	∞	zero	No	full	Post-D				✓	
Song and Chen	[186]	both	*	*	1	Manually	Excl.	No	n/a	∞	zero	No	full	Pre-D				✓	
Vahdani and Zandieh	[197][v]	both	n/a	1	1	Automated	Excl.	No	∞	∞	zero	No	full	Post-D				✓	
Vahdani *et al.*	[195][v]	both	n/a	1	1	Automated	Excl.	Yes	∞	∞	zero	No	full	Post-D				✓	
Vahdani *et al.*	[196]	both	n/a	1	1	Automated	Excl.	No	∞	∞	zero	No	full	Post-D				✓	
Yu and Egbelu	[220][v]	both	n/a	1	1	Automated	Excl.	No	∞	∞	zero	No	full	Dest.				✓	

Table 1.10: Truck sequencing

		Strategical level				Tactical level				Operational level				Performance measures						
		On which doors	Shape	Nb inbound doors	Nb outbound doors	Internal transport	Service mode	Pre-emption	Storage capacity	Resources capacity	Arrival time	Departure time	Truck filling	Interchangeability	Inventory level	Total product stay time	Truck time deviation	Loading time	Unloading time	Makespan	Balance workload
Álvarez Pérez *et al.*	[11]^{v}	both	*	*	*	ns	n/a	No	∞	∞	/truck	Both	full	Pre-D			✓				
Bellanger *et al.*	[21]	both	n/a	*	*	*	Excl.	No	0	∞	zero	No	full	Dest.						✓	
Berghman *et al.*	[22]	both	*	*	*	Manually	Mixed	No	∞	∞	/truck	Both	n/a	Pre-D		✓					
Boloori Arabani *et al.*	[24]^{v}	both	n/a	1	1	Automated	Excl.	No	∞	∞	zero	Outb.	n/a	Post-D			✓				
Boloori Arabani *et al.*	[25]^{v}	both	n/a	1	1	Automated	Excl.	No	∞	∞	zero	No	n/a	Dest.						✓	✓
Boloori Arabani *et al.*	[26]	both	n/a	1	1	*	Excl.	No	0	∞	zero	Outb.	n/a	Post-D			✓			✓	
Boloori Arabani *et al.*	[27]	both	n/a	1	1	*	Excl.	No	0	∞	zero	Outb.	full	Pre-D			✓			✓	
Boysen *et al.*	[33]^{v}	both	n/a	1	1	*	Excl.	No	∞	∞	zero	No	full	Post-D						✓	
Boysen	[28]^{v}	both	*	*	*	Manually	Excl.	No	0	∞	zero	Outb.		Pre-D		✓	✓	✓			
Chen *et al.*	[44]	both	*	*	1	ns	Excl.	No	∞	∞	zero	No	n/a	Pre-D						✓	
Li *et al.*	[122]^{v}	both	*	*	*	ns	n/a	No	∞	∞	/truck	Both	n/a	Pre-D			✓				

Table 1.11: Truck scheduling

1 near Annecy). For confidentiality reasons, the names of the companies and of the platforms have been made anonymous. In each platform, we were able to observe the ongoing operations and interview the platform manager and/or the logistics director of the company. The interviews were all carried out with the same interview grid, developed in order to be able to compare the on-field observations with the literature review described in the previous section. The interview grid (in French) is available in Appendix A.

The eight platforms are very diverse in terms of size, products manipulated, and activity volumes. We believe they make a good sample of the reality of cross-docking platforms in France.

The eight visited platforms are a good sample of French crossdocks.

Table 1.12 summarizes our observations and the outcomes of our interviews with the platform managers. Platforms B, F, G and K handle various retail products (electronics, cosmetics, clothes, toys...); platforms S and T respectively handle fresh food and frozen food; platforms C and Y deal with parcel delivery. Platforms B, C, G and K belong to the companies selling the products, whereas the other platforms are logistic service providers carrying their clients' products. The first section of the table gathers information that give an idea of the platform size: its physical surface, but also its number of employees and yearly/daily volumes. Platform K, that unloads bulk containers only (about 800 per year), could not give an estimation in cases. Platform Y crossdocks parcels only during peak times (*e. g.* Christmas) or punctually when its client its saturated; therefore yearly estimations would not have made sense in their case.

Half of the platforms visited carried out pure cross-docking operations, where all products stay less than 24 hours in the platform. In all four cases, all products received on a given day should leave the same day; absolutely no product is stored overnight. Those four platforms are the ones dealing with food and parcel deliveries – two sectors where the flow of goods must be extremely fast. The other four are either holding their crossdocked items in retention for up to three days before loading them in the outbound trucks, or storing some of the received items in racks for a longer period of time.

At the end of the interview with the platform managers, each was asked to state the main issues or needs at the moment. Their answers are summarized in Table 1.13.

1.2.4 *Discussion*

In truck sequencing and truck scheduling problems, 23 out of the 38 papers studied work on an imaginary crossdock with one inbound and one outbound door. It is not too surprising to observe that real-life platforms have more than two doors – 35 on the average for our sample of eight platforms. In this section we discuss other gaps observed between the literature and our observations in industry, or in

	Size information						Strategical level					Tactical level				Operational level				Performance measures									
	Surface (m²)	Nb permanent employees	Flow quantification [4]	Cases [5] handled per year	Cases handled per day	Trucks handled per day	Shape	Total number of doors	Nb inbound doors	Nb outbound doors	Internal transport	Service mode	Preemption	Storage capacity	Resources capacity	Arrival time	Departure time	Truck filling	Interchangeability	Inventory level	Working hours	Congestion	Number of touches	Truck punctuality	Loading time	Unloading time	Door utilization	Products not loaded	Makespan
B	2 500	10	small	60 000	300	10	I	4			Combination	Mixed	no	lim	lim	/truck	Outb.	full	Post-D		✓								
C	30 000	230	big	261 000 000	180 000	320	I	64	32	32	Automated	Dest.	no	lim	lim	0	Both	full	Dest.										✓
F	35 000	280	medium/big	11 700 000	35 000	150-200	I	52			Manually	Mixed	no	lim	lim	/truck	Both	full	Post-D.	✓		✓		✓			✓		✓
G	46 000	130	medium	24 000 000	540	50	I	34	14	20	Combination	Dest.	no	∞ [6]	lim	/truck	Outb.	full	Dest.		✓	✓				✓		✓	
K	24 000	57	medium	800 cont.	nc	10-60	I	32			Manually	Mixed	no	lim	lim	/truck	Both	full	Post-D.		✓								
S	14 000	170	medium	50 300 000	220 000	105	L	40			Manually	Mixed	no	lim	lim	/truck	Both	full	Post-D.		✓		✓				✓		
T	4 550	27	medium	13 725 000	45 000	70	I	22			Manually	Mixed+Dest. [7]	no	lim	lim	0	Both	full	Dest.		✓		✓						
Y	34 000	10	big	n/a	100 000	100	I	30	10	20	Manually	Excl.	no	lim	lim	0	Both	full	Dest.		✓	✓							✓

Table 1.12: On-field observations

[4] As perceived by the manager

[5] Case: describes a unit of measure and the way multiple physical units are packaged. A case would typically be a sealed corrugated carton where a standardized quantity (greater than one) of a specific item is packed. Definition from CSCMP [52].

[6] Of course this platform does not have a real infinite storage capacity; but its capacity is big enough that it is not considered as a possible constraint.

[7] The destinations are allocated to specific doors that can also be used for inbound trucks.

	Main concerns
B	Schedule the human resources needed for scan and packaging operations
C	*No issues mentioned*
F	Problems for forecasting the volume of activity, thus the number of employees needed
G	Forecasts known at the last minute: employee timetabling is a delicate task
K	Managing the delayed trucks
S	When and at which door to schedule the outbound trucks
T	How to absorb delayed trucks during peaks of activity; how to schedule the employees (done manually)
Y	How many temporary workers are needed and for how long

Table 1.13: Main managerial issues in the visited platforms

the contrary we point out elements on which researchers and practitioners converge.

SHAPE. When a shape is mentioned in the literature, it is almost always an I-shape. Seven out of the eight platforms visited were also I-shaped, so this common assumption seems to be justified. All visited platforms have less than 100 doors, a size for which an I-shape is more efficient according to Bartholdi and Gue [19].

INTERNAL TRANSPORT. 53% of the papers in the literature review assume a manual internal transport, 15% an automated one and 1% a combination of both (the remaining papers do not state the type of internal transport used). These results match quite well our observations on a small sample of real platforms.

SERVICE MODE. Four of the visited platforms have a destination-exclusive or exclusive mode of service (in which case we state the number of inbound and outbound doors), while five have a mixed mode of service, *i.e.* use the same doors for inbound and outbound operations (the total does not amount to eight because one platform uses a mixed mode of service but the doors are dedicated to destinations in their outbound mode). An exclusive mode of service may lower the efficiency of the dock utilization, but is still widely used because having fixed inbound and outbound doors eases the operation management inside a platform.

Academic works also use both assumptions (6% of all papers use a destination-exclusive mode of service, 68% an exclusive mode, 13% a mixed mode), but all the papers on truck sequencing and most of those on truck scheduling assume an exclusive mode of service. Because the number of platforms using a mixed mode is not negligible in practice, the community should consider investigating truck sequencing and scheduling in a mixed mode of service.

PREEMPTION. None of the platforms studied use preemption when unloading or loading their trucks – and the question raised real surprise among the managers, who wondered what would be the point of interrupting a loading or unloading operation. Preemption is not used much in the literature, and most of the time in problems with only one inbound and one outbound door. The assumption might be valid in a production crossdock (where the inbound side is a conveyor from a production line, for example), but does not seem to be a common practice for a regular crossdock.

STORAGE AND RESOURCE CAPACITY. The idea that resource and storage capacity are limited in most of the real-world logistic platforms is quite straightforward, and comforted by our observations. However this double constraint is taken into account in only 3% of the cross-docking literature, and not at all in the articles dealing with truck sequencing or truck scheduling. This is an important gap between theory and practice, that needs to be filled by including such constraints in the theoretical models.

We also observe, from the list of managerial issues in Table 1.13, that knowing the number of employees needed and scheduling them is a major concern among the managers. The uncertainty of the activity volumes makes scheduling a difficult task. We found almost no mention of this question in the cross-docking literature.

ARRIVAL AND DEPARTURE TIME. Truck arrivals in platforms C, T and Y are concentrated, which means that all trucks arrive almost at the same time. This type of organization is strongly linked to the sort of products handled, parcels and frozen food, that need to be sorted and dispatched in a time window as short as possible. Note that the number of trucks handled by these platforms is quite big, which can cause an important congestion in the parking lot or even the surroundings of the platform. Other platforms handling retail products deal with trucks arriving rather regularly through the day – in general the arrival time of each truck is known quite precisely when the platform is managed by the company that owns the products, and is often unknown by logistic service providers. The management of delayed arrivals is also a point appearing twice in our list of main managerial issues (Table 1.13).

Both situations, concentrated and scattered arrival times, are almost equally studied in the literature; but the case of uncertain arrival times is only addressed by ten papers: Acar *et al.* [2], Baptiste and Maknoon [16], Guignard and Hahn [92], Konur and Golias [112, 113], Li *et al.* [122, 127], Shakeri [178], Shakeri *et al.* [182], Werners and Wülfing [214].

Departure times are imposed for the inbound and outbound trucks in most platforms – the inbound truck departure times are unconstrained for platform B and G only, who both receive products shipped by their own company. Departure times are not constrained at all in 48% of the articles of our literature review: this assumption does not seem to hold in real life.

TRUCK FILLING. Truck filling is a subject on which researchers and practitioners seem to reach the general consensus that trucks should be fully loaded. Note that it means that trucks should be loaded with all the pallets, parcels or items that were planned to be loaded in: it does not necessarily mean that the available space in each truck is fully used – as a matter of fact, this is rarely the case.

INTERCHANGEABILITY. We observed an equal number of crossdocks where each product is dedicated to a specific outbound truck, and crossdocks where each product is headed to a destination and the exact product/truck allocation is decided when loading the outbound trucks. This also matches the assumptions commonly made by researchers.

PERFORMANCE MEASURES. The makespan and the distance traveled by workers are prominent performance measures in our literature review; the managerial practices do not reflect the same tendencies. Reducing the distance traveled is admittedly among the manager's preoccupations for productivity and ergonomic reasons. However, they do not cite the distance traveled as a performance measure because it is not an easily accessible data for them.

The success of the makespan as a performance measure in mathematical models is easily explained by its popularity in the scheduling field in general. However, finishing early (*i.e.* minimizing the makespan) is considered as important by only three of the managers we interviewed: oftentimes the end of the day depends on the departure time of the last truck, which is not necessarily flexible. The important measure for six of the eight platforms considered is the number of hours worked by the employees of the platform, which is due to the fact that most of the work is carried out manually. Surprisingly enough, congestion is a major concern in the larger platforms only: it seems that the bigger the platform, the bigger the flows and risks of congestion.

1.2.5 *Conclusion*

We used the comparison framework described in section 1.2.1 to compare the literature review on the one hand, and the practices observed in industry on the other hand. We observed some gaps between theory and practice, that would need to be filled by focusing on the following research areas:

- truck sequencing and scheduling with a mixed service mode, *i. e.* when doors can serve as inbound or outbound doors as needed;
- including storage capacity and resource capacity in cross-docking models;
- scheduling/timetabling of cross-dock employees;
- scheduling operations under uncertain or late truck arrivals;
- considering the number of hours worked by the platform employees as an objective function in cross-docking models.

This literature review was conducted on articles dealing with cross-docking operations. However, other fields investigate problems that can be closely related to cross-docking operations problems. It is *e. g.* the case of railway yards management problems, that are reviewed by Boysen *et al.* [32], and port operations reviewed by Vis and de Koster [207]. Flight gate scheduling problems in airports are also closely related to truck-to-door scheduling problems: a state-of-the art can be found in Dorndorf *et al.* [64].

1.3 PROBLEM AND MOTIVATIONS

The study carried out in the previous section helps drawing the objectives and motivations for the rest of this dissertation.

1.3.1 *Objectives of the study*

Our objective in this dissertation is to fill the most critical of the gaps between literature and industry practice identified in section 1.2. Because they were mentioned several times in our discussions with the platform managers, we focus on two main issues:

MANAGEMENT OF LATE ARRIVALS. One of the managers' main concerns was to determine the best way to handle a delayed truck without disturbing the rest of the operations. The first objective of this dissertation is therefore to propose a scheduling tool that would help managers to handle late trucks with as few perturbations as possible for other ongoing operations. In order to be able to study truck scheduling under late arrivals, an intermediate objective is to build a deterministic truck scheduling model as a first step.

EMPLOYEES SCHEDULING/TIMETABLING. Scheduling the employees' working hours seems to be a hard and time-consuming task for the majority of the managers we have met; yet the timetabling process needs to be quick to be adaptable in case of changes in the available information regarding the activity volume. The second objective of this dissertation is therefore to propose a decision-support tool for employee timetabling.

Late arrivals cannot be handled without available manpower, thus both issues seem to be strongly linked. A last objective is therefore to study the links between the two problems, in order to propose an integrated solution.

1.3.2 *Problem statement*

From the objectives detailed above, we derive three questions that this dissertation aims at answering:

How to schedule truck and pallet flows in a cross-docking platform?

How to manage delayed trucks without disturbing other ongoing operations?

How to schedule the workers in a way that fits the operations workload?

1.3.3 *Dissertation outline*

The rest of the document is organized in three different parts.

The first part, composed of chapter 2, chapter 3 and chapter 4, addresses the truck scheduling problem. Chapter 2 deals with the deterministic case and therefore answers the first of the three questions of section 1.3.2. Handling delayed trucks without disturbing the ongoing operations is possible if the trucks schedule is robust to truck delays; therefore chapter 3 proposes a methodology and a set of metrics to evaluate the robustness of the model – when the truck arrivals and departures are subject to uncertainties, but also when the transfer and unloading times are variable. These robustness metrics are then used in chapter 4 to compare various robust variations of the original model and thus answer the second question.

In the second part, composed of chapter 5, the employee scheduling problem is addressed on different time scales (weekly timetabling and daily rostering) in order to answer the third question.

Finally, chapter 6 explains how the truck scheduling model and the employee scheduling model can be used together.

The conclusions at the end of each chapter are technical ones; the global conclusion and perspectives are given at the end of the document.

In preparing for battle I have
always found that plans are useless,
but planning is indispensable.

— Dwight D. Eisenhower

Chapter 2

OPTIMIZING CROSSDOCK TRUCK SCHEDULING

This chapter aims at proposing a decision-support tool to schedule truck arrivals/departures and pallet transfers (including storage) in a cross-docking platform. It is assumed that the manager who schedules the operations of a given day knows the list of inbound and outbound trucks planned on that day and their content. Transportation providers use a reservation system to give their preferred arrival and departure times. The objective is to schedule the trucks and pallet transfers, minimizing the number of pallets temporarily stored and maximizing the transportation providers' satisfaction regarding the presence time windows that are allocated to each truck in the final schedule. This chapter proposes to model the problem with an Integer Program (IP) and to solve it with three different heuristics.

The work presented in this chapter is also presented in the following articles:

Ladier, A.-L., and Alpan, G. 2013. Scheduling truck arrivals and departures in a crossdock: earliness, tardiness and storage policies. In *International Conference on Industrial Engineering and Systems Management*. Rabat, Marocco.

Ladier, A.-L., and Alpan, G. Crossdock truck scheduling with time windows: Earliness, tardiness and storage policies. Submitted for publication in the *Journal of Intelligent Manufacturing*.

OPTIMISATION DE LA PLANIFICATION DES CAMIONS

Ce chapitre vise à proposer un outil d'aide à la décision pour planifier les opérations (arrivées et départs des camions, transfert de palettes) d'une plateforme de cross-docking. Le manager qui planifie les opérations d'un jour donné dispose de la liste des camions prévus en entrée et sortie, ainsi que de leur contenu. Grâce à un système de réservation, les transporteurs indiquent leurs plages horaires préférées pour chacun des camions. Si nécessaire, il est possible de planifier un camion à une plage horaire différente que celle qu'il a demandée ; mais cette situation doit être évitée autant que possible, car elle risque de perturber la tournée du transporteur. Si le camion de sortie correspondant n'est pas à quai au moment de traiter une palette entrante, celle-ci est temporairement placée en stock. Comme l'opération va demander deux coups de fourche du cariste au lieu d'un (soit deux fois plus de ressources), on cherche également à minimiser ces situations. L'objectif est donc de planifier les camions et les transferts de palettes de façon à minimiser la quantité de palettes mises en stock, et à maximiser la satisfaction des transporteurs concernant les plages horaires qui leur sont attribuées dans le planning final. Dans le cas déterministe, on modélise le problème par un programme linéaire en nombres entiers (PLNE). Les variables de décisions concernent d'une part la plage horaire attribuée à chacun des camions (entrants et sortants), et d'autre part les mouvements de palettes (à chaque unité de temps, le nombre de palettes déplacées d'un camion à l'autre, depuis et vers le stock). Le PLNE ainsi formulé (IP*) n'est utilisable que pour de très petites instances. Afin de pouvoir traiter des instances de taille réaliste, nous proposons trois heuristiques. Les deux premières décomposent le problème en deux PLNE plus petits. En supposant que les plages horaires exprimées pour les camions sortants sont les plages horaires définitives, le PLNE IP1 détermine le planning des camions sortants. Ce planning est utilisé comme une donnée d'entrée pour une version relaxée d'IP*. La seconde heuristique suit la même logique, en fixant cette fois les camions entrants. La troisième heuristique est une recherche tabou, qui détermine la valeur de la composante "stock" de la fonction objectif en résolvant un problème de flot maximum. Les performances des trois heuristiques sont testées et comparées dans différents cas de figure.

2 OPTIMIZING CROSSDOCK TRUCK SCHEDULING

As shown in chapter 1, the punctuality of the trucks is of crucial importance for the platform managers. This chapter therefore aims at providing logistic managers with a decision-support tool to schedule the truck-related operations as well as the storage plan.

After describing the problem in section 2.1, a first formulation using an Integer Programming model (section 2.2) is detailed and tested. In order to overcome the computational limitations, three heuristics are proposed in section 2.3.

2.1 TRUCK SCHEDULING WITH TIME WINDOWS: PROBLEM DESCRIPTION

The question is to plan inbound and outbound truck arrivals and departures as well as pallet moves through a crossdock. It is assumed that the platform manager knows the preferences of the transportation providers regarding arrival and departure times, for both inbound and outbound trucks. The resulting schedule should maximize the transportation providers' preference satisfaction and minimize storage.

In order to ensure the synchronization between inbound and outbound flows, it is also important to track the pallet moves. This information is valuable for the manager since it provides detailed information about the workload inside the platform, both for moves from trucks to trucks and for moves to and from storage.

2.1.1 *Assumptions*

According to the classification proposed in chapter 1 (section 1.2.2), the issue is addressed as a truck scheduling problem. Therefore, the spatial dimension is not taken into account. Unloading, scanning, transfer and loading operations are all done within the same time period; consequently the time period is defined to be long enough (*e.g.* at least half an hour) to ensure the product transfers in masked time.

The exact contents of the inbound trucks (number of pallets for each destination) are assumed to be known. When a truck arrives, it is entirely unloaded on the dock, and the pallets can then be picked from the dock in any order.

The doors have an *exclusive* mode of service. No preemption is allowed. The storage capacity is supposed to be unlimited. The outbound trucks have a fixed capacity F, and cannot leave before they are fully loaded. The resource capacity is limited by the number of workers present, the number of material handling equipment or both: no more than M units can be moved in one time period inside the platform.

When the required outbound truck is not available to load a given pallet, the pallet is placed in storage. All pallets entering the platform on a given day will leave it on the same day, so the pallets are not stored for a long time. Therefore, the model does not follow a FIFO policy to empty the stock, and the pallets in storage can be taken out in any order. Since the items are stored for a short amount of time, the holding costs are negligible compared to the cost of extra handling. Placing an item in the temporary storage area is more costly than directly transferring it from an inbound to an outbound truck, since the item is touched twice instead of just once. Therefore, the goal is to minimize the total number of products put in storage.

The transportation provider expresses his preferences about the wished arrival and departure times for all trucks, *i. e.* a preferred time window of presence. Another goal is to maximize the transportation provider's satisfaction: a time window will be penalized if it starts before, or ends after, the wished arrival or departure time. Hence, both the earliness and tardiness of the inbound and the outbound trucks are considered. This objective enters into the "truck time deviation" performance measure in the comparison framework proposed in section 1.2.1.

The different assumptions for the problem considered are summarized in Table 2.1.

	Strategical level				Tactical level				Operational level				Perf measures	
On which doors	Shape	Nb inbound doors	Nb outbound doors	Internal transport	Service mode	Pre-emption	Storage capacity	Resources capacity	Arrival time	Departure time	Truck filling	Interchangeability	Inventory level	Truck time deviation
both	*	*	*	Manually	Exclusive	No	∞	lim	/truck	Both	full	Dest.	✓	✓

Table 2.1: Classification of the truck scheduling problem studied in chapter 2

2.1.2 *Similar problems in the literature*

Time windows are introduced by Li *et al.* [122] for the inbound trucks in the truck sequencing problem: each inbound truck ("incoming container") has a release time and due date, while each outbound truck has only a due date. The goal is to minimize earliness and tardiness penalties, *i. e.* the absolute value of the difference between the actual truck departure time and its due date. The problem is modeled as a machine scheduling problem and solved with two heuristics (squeaky wheel optimization and linear programming, both embedded in a genetic algorithm). Álvarez Pérez *et al.* [11] propose to combine two metaheuristics (reactive GRASP and tabu search) to solve the same problem.

GRASP stands for Greedy Randomized Adaptive Search Procedure.

Golias *et al.* [81] propose to further extend these work into a truck-to-door sequencing problem, by adding another objective: maximizing the total throughput of the platform. This is actually done by minimizing the total service time of all trucks. Early and late truck departures at both the inbound and outbound doors are also penalized when the departure is outside a predefined time window. Our approach also makes use of time windows, but considers that both the arrival and departure of the trucks should be in the time window, since the arrival time is also of importance for the transportation provider.

As can be seen in Table 1.11, another work that uses truck time deviation as a performance measure in a crossdock truck scheduling problem is by Boysen [28] and focuses on the outbound trucks only. Boysen proposes a model for a frozen food platform in which the storage is forbidden. The objective is to minimize the flow time, processing time and tardiness of the outbound trucks.

In a logistics platform, the punctuality of the trucks is of crucial importance for the managers, not only for the outbound but also for the inbound trucks. Early truck arrivals may disturb the internal operations as much as delays (*e. g.* unexpected congestion inside the platform or in the parking area, need for a reorganization of internal resources, *etc.*). Therefore, unlike previous work found in the literature, this chapter considers both the earliness and tardiness of the trucks, for both inbound and outbound operations.

2.1.3 *Input data*

As a convention in the entire document, we use calligraphic letters to name the sets, capital letters for the known input parameters, and lower-case letters for decision variables.

The input data and the decision variables are defined over the following sets:

- $\mathcal{H}$ set of time periods (*e. g.* half an hour) in the planning horizon considered;
- $\mathcal{I}$ set of inbound trucks;
- $\mathcal{O}$ set of outbound trucks;
- $\mathcal{C}$ set of clients to whom the pallets should be delivered.

From the assumptions detailed in section 2.1.1, the input data considered include:

Q_{ic} number of pallets for client $c \in \mathcal{C}$ in truck $i \in \mathcal{I}$;
Z_{co} $= 1$ if truck $o \in \mathcal{O}$ is for client $c \in \mathcal{C}$, 0 otherwise;
N^I number of inbound doors;
N^O number of outbound doors;
M maximum number of pallets that can be moved during one time period inside the platform;
F number of pallets needed to fully load one outbound truck.

The data listed above correspond to strategical decisions (physical constraints in the crossdock) or tactical decisions (destinations and capacity of different trucks). At the operational level, those decisions are constraints that cannot be violated. The model, therefore, incorporates them as hard constraints. The only data corresponding to an operational decision is M, the internal capacity of the platform. The value of M can depend on the available material handling equipment, and on the number of employees present on the day considered. In this chapter we consider that M has a fixed value for the whole planning horizon; in chapter 6 this assumption will be relaxed by varying M through the day, to incorporate workers' timetable.

The earliest possible arrival time and latest possible departure time of each inbound (resp. outbound) truck are known. In the general case, they correspond to the beginning and the end of the planning horizon – however, some hard constraints expressed by the transportation provider can also be taken into account through this data. The wishes of the transportation providers are known regarding the arrival and presence time of trucks: the objective is to satisfy them as much as possible.

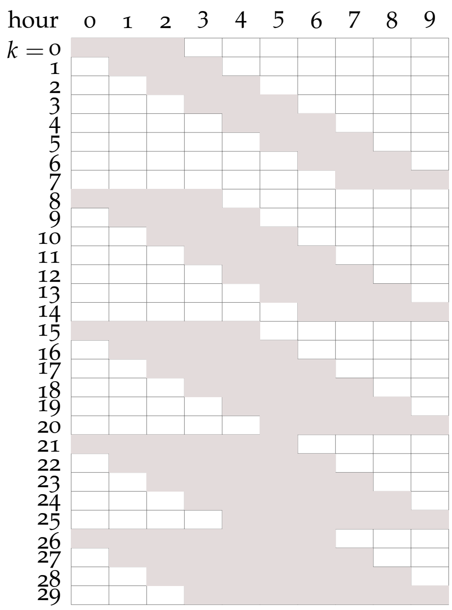


Figure 2.1: Example of matrix W^I for a given truck i
(for slot length from 3 to 7 hours)

2.2 FIRST FORMULATION: INTEGER PROGRAMMING MODEL

The problem presented in section 2.1.1 can be formulated as an Integer Programming (IP) model. This section presents and explains the IP model, the complexity of the problem, and numerical experiments.

2.2.1 *Integer program*

The model aims at defining the truck schedule, with the objective of being as close as possible to the wishes of the transportation providers, and at the same time minimizing the storage.

Some modeling choices have to be made regarding the definition of the truck presence slots and their penalties. The earliest possible arrival time and latest possible departure time being known, the possible presence slots of a given truck can be enumerated. We note $\mathcal{K}_i$ (resp. $\mathcal{K}_o$) as the set of possible presence slots of the truck $i \in \mathcal{I}$ (resp. $o \in \mathcal{O}$). These possible presence slots are described by matrices W^I and W^O, where:

W^I_{ikh} = 1 if hour $h \in \mathcal{H}$ is in slot $k \in \mathcal{K}_i$ for inbound truck $i \in \mathcal{I}$;
W^O_{okh} = 1 if hour $h \in \mathcal{H}$ is in slot $k \in \mathcal{K}_o$ for outbound truck $o \in \mathcal{O}$.

An example of matrix W^I is given in Figure 2.1.

The wishes of the transportation providers are seen as soft constraints: if trucks are scheduled outside their wished slots, penalties are paid. Those penalties P^I and P^O are therefore defined as follows:

P^I_{ik} penalty paid for using slot $k \in \mathcal{K}_i$ for truck $i \in \mathcal{I}$, if k is different from the wished time window expressed by the transportation provider;
P^O_{ok} penalty paid for using slot $k \in \mathcal{K}_o$ for truck $o \in \mathcal{O}$.

This way to define the penalties enables one to use any cost structure to penalize different slots.

Monitoring the pallet moves is necessary to ensure the synchronization of the inbound and outbound flows. The model therefore uses the following decisions variables, that are summarized in Figure 2.2:

x_{hio} amount of pallets transferred from inbound truck $i \in \mathcal{I}$ to outbound truck $o \in \mathcal{O}$ at time period $h \in \mathcal{H}$;
w^I_{ik} =1 if slot $k \in \mathcal{K}_i$ is chosen for truck $i \in \mathcal{I}$, 0 otherwise;
w^O_{ok} =1 if slot $k \in \mathcal{K}_o$ is chosen for truck $o \in \mathcal{O}$, 0 otherwise;
s^I_{hic} amount of pallets for client $c \in \mathcal{C}$ going from truck $i \in \mathcal{I}$ to the storage location at time period $h \in \mathcal{H}$;
s^O_{ho} amount of pallets going from the storage location to truck $o \in \mathcal{O}$ at time period $h \in \mathcal{H}$;
s_{hc} amount of pallets for client $c \in \mathcal{C}$ stored at time period $h \in \mathcal{H}$.

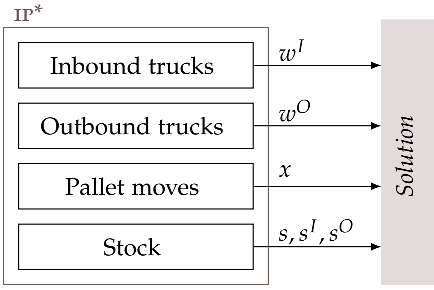


Figure 2.2: Outputs of the Integer Program IP*

The planning problem can now be formulated as an Integer Program noted IP* – see below.

The objective is to minimize the time window penalties for inbound and outbound trucks defined by constraints (1) and (2), as well as the number of pallets placed in storage defined by constraint (3). α_0, β_0 and γ_0 are coefficients weighting those often conflicting objectives.

Constraint set (4) (resp. (5)) checks that the number of inbound (resp. outbound) trucks present during a given time period does not exceed the number of inbound (resp. outbound) doors.

Constraint set (6) (resp. (7)) ensures that the pallet moves from inbound trucks (resp. to outbound trucks) occur only when the con-

$$\min \quad \alpha_0 \Pi_0^\alpha + \beta_0 \Pi_0^\beta + \gamma_0 \Pi_0^\gamma$$

$$
\begin{array}{llll}
\text{s.t.} & \Pi_0^\alpha = \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} P^I_{ik} w^I_{ik} & & (1) \\
& \Pi_0^\beta = \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} P^O_{ok} w^O_{ok} & & (2) \\
& \Pi_0^\gamma = \sum_{h\in\mathcal{H}, i\in\mathcal{I}, c\in\mathcal{C}} s^I_{hic} & & (3) \\
& \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} W^I_{ikh} w^I_{ik} \le N^I & \forall h\in\mathcal{H} & (4) \\
& \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} W^O_{okh} w^O_{ok} \le N^O & \forall h\in\mathcal{H} & (5) \\
& x_{hio} + s^I_{hic} \le F \sum_{k\in\mathcal{K}_i} W^I_{ikh} w^I_{ik} & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (6) \\
& x_{hio} + s^O_{ho} \le F \sum_{k\in\mathcal{K}_o} W^O_{okh} w^O_{ok} & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (7) \\
& \sum_{h\in\mathcal{H}, o\in\mathcal{O}} Z_{co} x_{hio} + \sum_{h\in\mathcal{H}} s^I_{hic} = Q_{ic} & \forall i\in\mathcal{I}, c\in\mathcal{C} & (8) \\
& \sum_{i\in\mathcal{I}, h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s^O_{ho} = F & \forall o\in\mathcal{O} & (9) \\
& \sum_{o\in O} x_{hio} + \sum_{c\in\mathcal{C}} s^I_{hid} \le M & \forall i\in\mathcal{I}, h\in\mathcal{H} & (10) \\
& \sum_{k\in\mathcal{K}_i} w^I_{ik} = 1 & \forall i\in\mathcal{I} & (11) \\
& \sum_{k\in\mathcal{K}_o} w^O_{ok} = 1 & \forall o\in\mathcal{O} & (12) \\
& s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s^I_{hic} - \sum_{o\in\mathcal{O}} Z_{co} s^O_{ho} & \forall c\in\mathcal{C}, h\in\mathcal{H}\setminus\{0\} & (13) \\
& s_{0c} = \sum_{i\in\mathcal{I}} s^I_{0ic} - \sum_{o\in\mathcal{O}} Z_{co} s^O_{0o} & \forall c\in\mathcal{C} & (14) \\
& & & \\
& x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc} \in \mathbb{N}^+ & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}, c\in\mathcal{C} & \\
& w^I_{ik}, w^O_{ok} \in \{0,1\} & \forall i\in\mathcal{I}, o\in\mathcal{O}, k\in K &
\end{array}
$$

IP*

cerned truck is present. Constraint set (8) makes sure that all pallets from a given inbound truck are unloaded and dispatched to the right client.

Constraint set (9) indicates the capacity of outbound trucks, and makes sure that they are fully loaded. Constraint set (10) limits the quantity of pallets transferred inside the platform at each time period.

Constraint sets (11) and (12) make sure that each inbound (resp. outbound) truck is assigned to a single presence time window. Constraint sets (13) and (14) give the stock conservation rule for all $h \in \mathcal{H} \setminus \{0\}$ and for $h = 0$, respectively.

2.2.2 *Complexity*

In this section, the considered problem is shown to be NP-hard in the strong sense even in a simplified case. The NP-hardness is proved by a transformation from the 3-partition problem, which is NP-hard in the strong sense (see Garey and Johnson [77]). The idea of the proof is inspired by Sadykov [173].

3-PARTITION PROBLEM. Consider two integers B and n, and a set of $3n$ integers $r_1, r_2, ..., r_{3n}$ given such that

$$\begin{cases} \sum_{i=1}^{3n} r_i = Bn \\ \frac{B}{4} < r_i < \frac{B}{2} \qquad \forall i \end{cases}$$

The 3-partition problem consists in determining if the set $\{1, 2, ..., 3n\}$ can be partitioned into n subsets $\{A_1, A_2, ..., A_n\}$ such that

$$\sum_{i \in A_j} r_i = B \quad \forall j \in \{1, 2, ..., n\}$$

In other words, the problem is to divide $3n$ elements whose sum is Bn into n groups of sum B. If such a partition exists, each group (each subset A_j with $j \in \{1, 2, ..., n\}$) contains exactly 3 elements.

TRANSFORMATION INTO OUR TRUCK SCHEDULING PROBLEM.
Using the same notations used to describe the 3-partition problem, let us consider an instance with a time horizon of n time units ($|\mathcal{H}| = n$) in which there are:

- 3 inbound and 3 outbound doors ($N^I = N^O = 3$);
- two different clients ($|\mathcal{C}| = 2$) that will be called client 1 and client 2;
- the length of each possible time slot k ($k \in \mathcal{K}_{i \in \mathcal{I}}$ or $k \in \mathcal{K}_{o \in \mathcal{O}}$) is one time unit;
- the platform's internal capacity is not a constraint ($M = \infty$);
- $3n$ inbound trucks indexed by $i \in 1, 2, ..., 3n$, each containing:
  - 1 item for client 1;

  - $n + r_i$ items for client 2;
- $3n$ outbound trucks, among which:
  - $2n$ trucks are dedicated to client 1 and have a capacity 1;
  - n trucks are dedicated to client 2 and have a capacity $3n + B$.

Here the truck capacities are different from one another, which is not the case in our model. However, this is not a loss of generality since a truck capacity F can be reached by adding items for a third client up to F.

Proposition. There exists a 3-partition if and only if there exists a solution to the corresponding instance of our truck scheduling problem with $\Pi_0^\gamma \leq n$ (less than n items put in storage).

Proof. Necessity. Suppose there exists a 3-partition $\{A_1, A_2, ..., A_n\}$. The j-th subset A_j is composed of three elements that we note i_{j1}, i_{j2} and i_{j3}. Let us build a solution to the truck scheduling problem such that $\Pi_0^\gamma \leq n$. The $3n$ inbound trucks can be divided into n groups of three trucks using the 3-partition. The $3n$ outbound trucks can easily be divided into n groups of three trucks as well, each group containing two trucks for client 1 and one truck for client 2. Let us consider a solution in which the j-th group of inbound truck and the j-th group of outbound trucks ($j \in \{1, 2, ..., n\}$) are present at the platform during the same time unit j (see Figure 2.3).

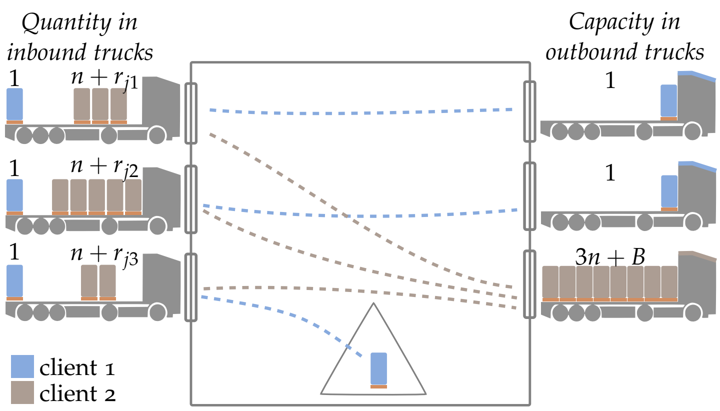


Figure 2.3: Solution for the considered instance at time j

Since $r_{j1} + r_{j2} + r_{j3} = B$ by definition of the 3-partition, there are exactly $3n + B$ pallets in the three inbound trucks: they can all be directly reloaded in the outbound truck dedicated to client 2. There are three pallets for client 1 to be unloaded in total: two can go directly to the corresponding outbound trucks, and one has to go to storage. Repeating the same pattern for the n time units of the horizon gives a solution in which n items are put in storage in total.

Sufficiency. Suppose that a solution to this instance of our truck scheduling problem exists, in which at most n products are put into storage; let us show that a 3-partition exists. Every outbound truck for client 2 needs $3n + B$ products, and no more than n can come from storage: thus it must be loaded with products coming from at least three different inbound trucks, that stay one time unit each. Besides the products for client 2, those three inbound truck contain necessarily three products from client 1. Since there are two available doors left, at most two products for client 1 can be directly reloaded: the others must go into storage. At most n products are put into storage, thus exactly one product per time unit goes into storage, and this product must be for client 1. Therefore, the three inbound trucks transfer all their products for client 2 $(3n + r_{j1} + r_{j2} + r_{j3})$ directly to the outbound truck of capacity $3n + B$, and fill it (the truck being present for only one time unit). This provides, for all time units $j \in \{1, 2, ..., n\}$, a partition of inbound trucks into triples $\{A_1, A_2, ..., A_n\}$ such that $\sum_{i \in A_j} r_i = B$. □

2.2.3 *Instance generation*

An instance generator has been developed, that takes as input values the parameters $|\mathcal{H}|$, $|\mathcal{I}|$, $|\mathcal{O}|$, $|\mathcal{C}|$, N^I, N^O, M, and F. From this basic data, it generates the rest of the data needed to fully express the problem. These data are generated based on random distributions, but ensuring that they stay feasible and consistent. For instance, each client should be served by at least one truck, and the inbound quantity for each client should be kept equal to the total capacity of the outbound trucks for this destination.

The instance generator always sets the earliest possible arrival time and latest possible departure time of each truck as the beginning and the end of the planning horizon. This corresponds to the most general case and does not restrict the solution space. Penalties P^I and P^O are directly calculated from W^I and W^O, as the number of hours outside the “wished” range in slot k (number of hours in blue in Figure 2.4).

The instance generator is available at www.g-scop.fr/~gaujalg/XDockInstances2, and details on the related algorithms can be found

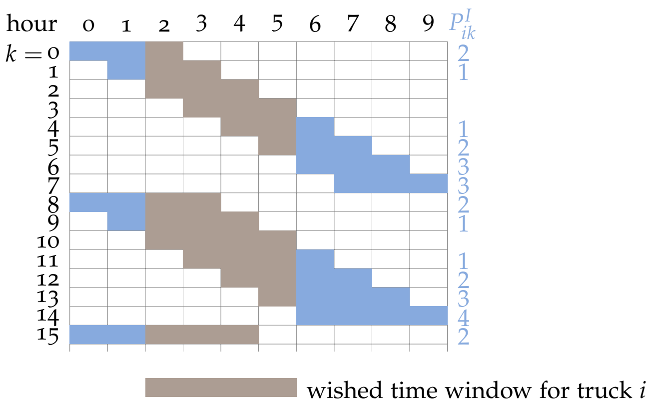


Figure 2.4: Extract of matrix W^I with penalties

in Appendix B. The instances used in this document are generated by this tool.

Since a standard truck can carry 33 European pallets, the inbound and outbound truck capacities F are set to 33 pallets in all instances. Three sets of instances are generated and named after their number of inbound and outbound doors: $N^I = N^O = 3$ for `set3+3`, $N^I = N^O = 12$ for `set12+12`, $N^I = N^O = 25$ for `set25+25`. They represent a very small platform (`set3+3`), a small platform (`set12+12`) and a medium-size one (`set25+25`) with different levels of activity. The other input parameters used to generate each instance are described in Table 2.2. The instances are named after their value of M.

`set3+3` - Instance name	17_1	17_2	17_3	17_4	17_5	34_1	34_2	34_3	34_4	34_5	34_6
number of hours	10	10	12	12	12	7	7	7	7	10	10
number of inbound trucks	5	5	6	6	6	5	5	6	6	7	7
number of clients	3	4	3	3	4	3	4	3	4	3	4
max pallets per hour M	17	17	17	17	17	34	34	34	34	34	34

`set12+12` - Instance name	85_1	85_2	85_3	85_4	85_5	102_1	102_2	102_3	102_4	102_5
number of hours	10	10	10	10	10	10	10	10	10	10
nb of inbound trucks	20	20	20	20	20	29	28	26	25	25
nb of clients	3	4	5	6	7	3	4	5	6	7
max pallets per hour M	85	85	85	85	85	102	102	102	102	102

`set25+25` - Instance name	255_1	255_2	255_3	255_4	255_5	272_1	272_2	272_3	272_4	272_5
number of hours	10	10	10	10	10	10	10	10	10	10
nb of inbound trucks	60	60	60	60	60	70	65	65	65	65
nb of clients	3	4	5	6	7	3	4	5	6	7
max pallets per hour M	255	255	255	255	255	272	272	272	272	272

Table 2.2: Description of the instances

An example of a complete instance is given in Figure 2.5 for instance 17_1. From the input data provided ($|\mathcal{H}| = 10$, $|\mathcal{I}| = |\mathcal{O}| = 5$, $|\mathcal{C}| = 3$, $N^I = N^O = 3$, $M = 17$, $F = 33$), the instance generator creates the other data detailed in Figure 2.5 and visualized in Figure 2.6. The contents of the inbound trucks correspond to Q_{ic} (Figure 2.5b), the color (client) of the outbound trucks are obtained from Z_{co} (Figure 2.5c), and the wished time windows match the data described in Figure 2.5a. The details of the data composing all other instances are available at `www.g-scop.fr/~gaujalg/XDockInstances2`.

	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$	$o=0$	$o=1$	$o=2$	$o=3$	$o=4$
wished arrival	4	0	6	1	5	5	3	1	7	0
wished departure	9	4	10	6	10	10	7	5	10	9

(a) Wished arrival and departure times for the trucks

	$c=0$	$c=1$	$c=2$
$i=0$	13	5	15
$i=1$	15	9	9
$i=2$	13	9	11
$i=3$	14	7	12
$i=4$	11	3	19

(b) Q_{ic}

	$o=0$	$o=1$	$o=2$	$o=3$	$o=4$
$c=0$	1	0	0	0	1
$c=1$	0	1	0	0	0
$c=2$	0	0	1	1	0

(c) Z_{co}

Figure 2.5: Detail of the data composing instance 17_1

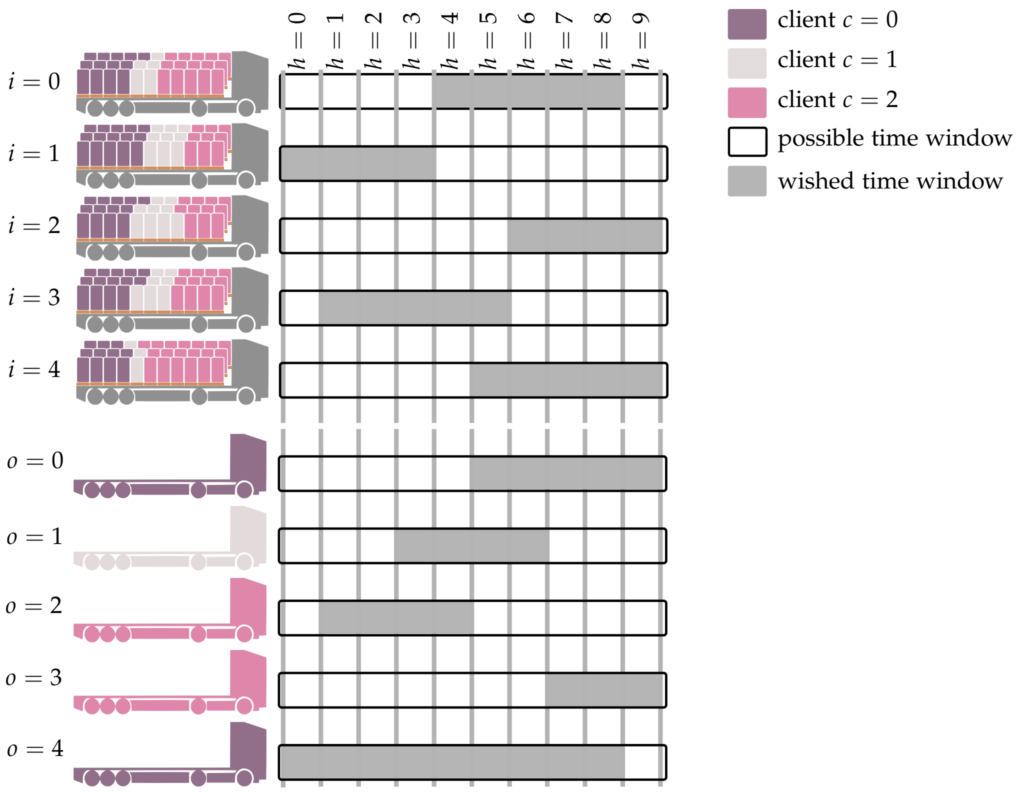


Figure 2.6: Visualization of instance 17_1

2.2.4 *Numerical results for the* IP *model*

In the entire document, all linear programs are run with IBM ILOG CPLEX Optimizers 12.2, on a personal computer with a 2.40 GHZ processor and a 4.00 GB RAM.

In this section only, we use very small instances to test the IP model. The input parameters used to create these instances with the instance generator described in section 2.2.3 are detailed in Table 2.3. Since there are only four to eight doors in total, the instances tested represent a very small platform.

$\lvert\mathcal{H}\rvert$	$\lvert\mathcal{C}\rvert$	M	F	$N^I = N^O$
10	4	4	4	2, 3 or 4

Table 2.3: Instance parameters to test IP*

As a first approximation, the coefficients α_0, β_0 and γ_0 are assumed equally important and are all set to 0.33.

The execution time of IP* is tested with different number of doors by simultaneously increasing the number of inbound trucks $\lvert\mathcal{I}\rvert$ and outbound trucks $\lvert\mathcal{O}\rvert$, keeping $\lvert\mathcal{I}\rvert = \lvert\mathcal{O}\rvert$. For the sake of comparison, the results are presented in Figure 2.7 as a function of the concentration of trucks. The concentration of trucks (in trucks per door per hour) is defined by the ratio:

$$R = \frac{\lvert\mathcal{I}\rvert + \lvert\mathcal{O}\rvert}{(N^I + N^O)\,\lvert\mathcal{H}\rvert} \tag{2.1}$$

$R \leq 1$

Even with very small platforms (8 doors or less) and low concentration rates, the execution times increases very quickly as shown in Figure 2.7. If ten seconds is considered the limit for a logistics manager to use this program as a daily decision-support tool, then we cannot deal with more than ten trucks on a platform with two inbound and

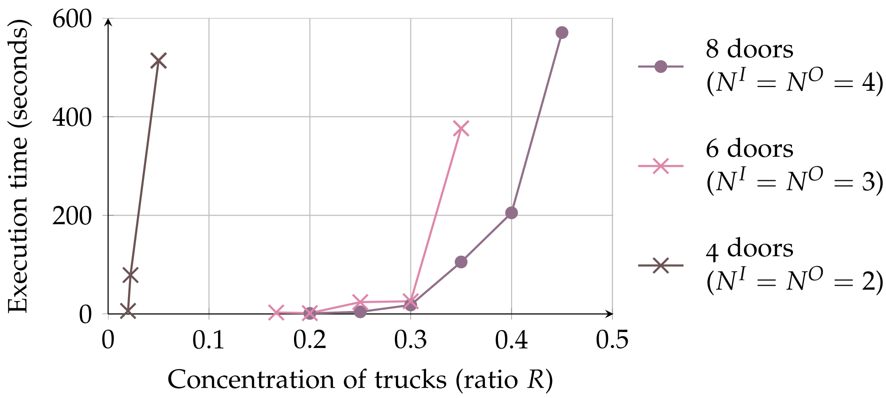


Figure 2.7: IP* execution time as a function of truck concentration

two outbound doors. Due to the complexity of the problem, the performances of IP* in terms of computation time do not permit to use it on a daily basis within crossdocks. It can solve the instances of `set3+3` (see the results in Table 2.4, and the visualization of the result of instance `17_1` in Figure 2.8) but cannot give a solution in a reasonable amount of time for the instances in `set12+12` and `set25+25`. In the next section, different heuristics are presented that can help overcome this issue.

	17_1	17_2	17_3	17_4	17_5	34_1	34_2	34_3	34_4	34_5	34_6
Π_0^α	0	1	0	0	0	0	0	0	0	0	0
Π_0^β	0	0	0	0	3	0	1	0	0	0	0
Π_0^γ	0	2	0	0	0	0	9	0	7	0	4
Exec. time (s)	0.288	0.201	0.234	0.231	0.956	0.087	5.149	0.164	4.512	0.506	0.559

Table 2.4: Results of IP* on instance `set3+3`

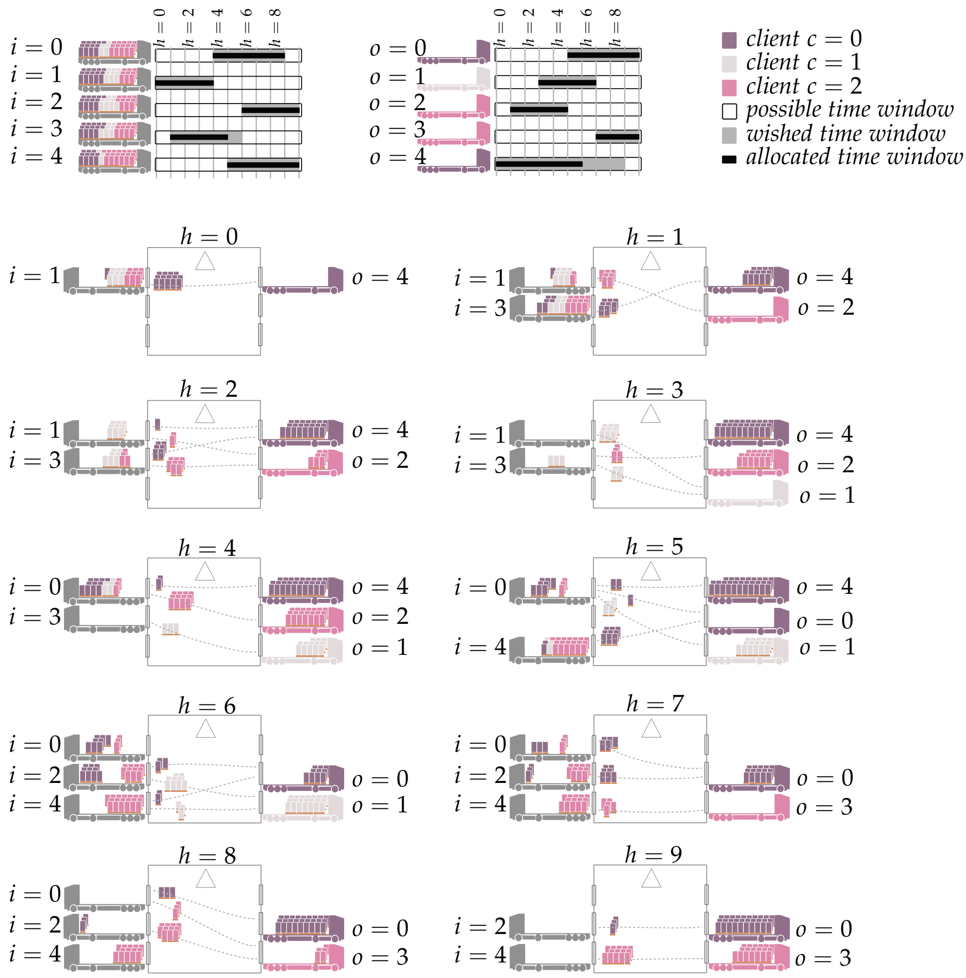


Figure 2.8: Visualization of the solution of instance `17_1`

2.3 HOW TO SCALE-UP: HEURISTICS

Since the Integer Program presented in the previous section takes too long to compute the instances of real-life size, we propose in this section three heuristics that can help solving the problem faster.

The integration of IP models in heuristics is usually called matheuristics.

The principle of the first two heuristics is to relax a part of IP*, in order to simplify the number of decisions taken during its execution. In the first heuristic (H1), the first step aims at obtaining an inbound trucks schedule used as data in a relaxed version of IP*, while the first step of heuristic H2 aims at calculating an outbound schedule. In both heuristics, the schedule of the first step is obtained by a dedicated integer program (IP1 or IP2). Heuristic 3 (H3) is a tabu search. Each iteration of the search fixes the schedule of both the inbound and the outbound sides, and an integer program IP^{*3} or a network flow evaluates the value of the objective function regarding the stock level. The principle is described in Figure 2.9.

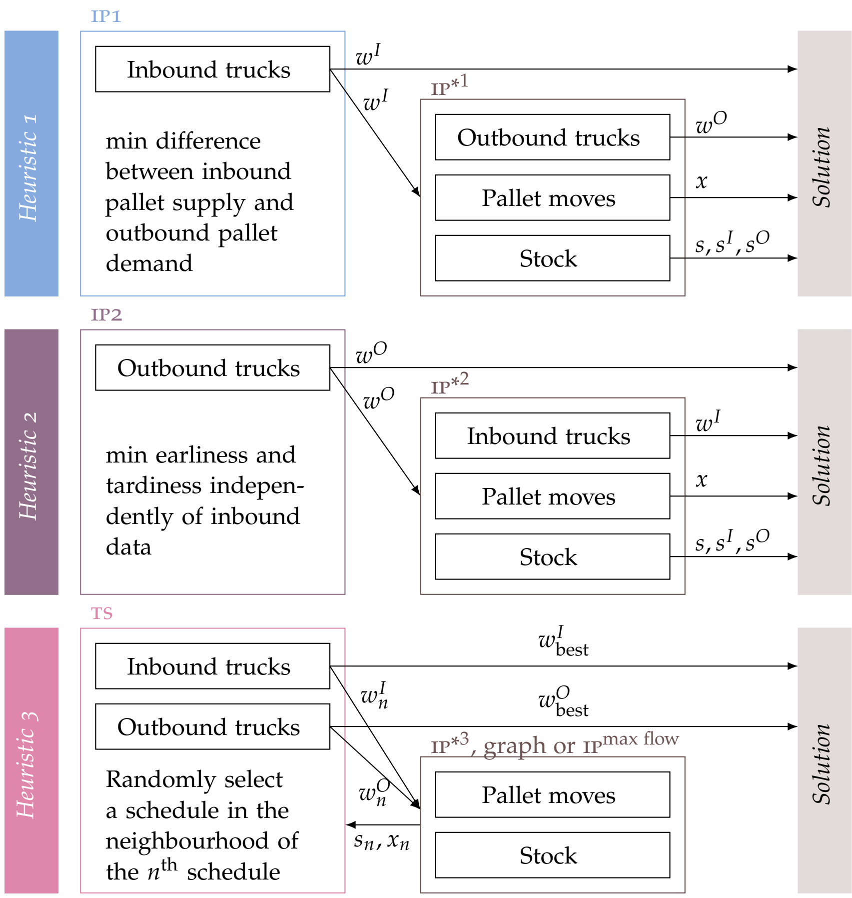


Figure 2.9: Principle of heuristics H1, H2 and H3

2.3.1 *Heuristic 1*

As the first step of heuristic H1, IP1 determine a good schedule for the inbound trucks using the wished presence time windows of the outbound trucks as data. Then the second step (IP^{*1}) uses the inbound truck schedule as data in order to compute the final schedule of the outbound trucks.

Let us assume, just for this first part of the heuristic, that the wished departure and arrival times of the outbound trucks are all satisfied. Using matrix Z which indicates the destination of each outbound truck, we can easily calculate X^O, a binary matrix defined as follows:

X^O_{ch} $= 1$ if there is an outbound truck for client c present at time period h, 0 otherwise.

Integer program IP1 uses w^I_{ik} as a decision variable, as well as two new variables that measure the difference between the inbound and the outbound plans:

δ^+_{ch} for time period $h \in \mathcal{H}$, positive difference between the number of pallets for client $c \in \mathcal{C}$ available to be unloaded, and the number of pallets that can be loaded in the trucks for client c present at the outbound doors.

δ^-_{ch} for time period $h \in \mathcal{H}$, negative difference between the number of pallets for client $c \in \mathcal{C}$ available to be unloaded, and the number of pallets that can be loaded in the trucks for client c present at the outbound doors.

IP1 is thus formulated as follows:

$$\min \quad \sum_{c\in\mathcal{C},h\in\mathcal{H}}(\delta^+_{ch}+\delta^-_{ch}) + \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_o} P^I_{ik} w^I_{ik}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} Q_{ic} W^I_{ikh} w^I_{ik} = M X^O_{ch} + \delta^+_{ch} - \delta^-_{ch} \qquad \forall c\in\mathcal{C}, h\in\mathcal{H} \qquad (15)$$

$$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} W^I_{ikh} w^I_{ik} \le N^I \qquad \forall h\in\mathcal{H} \qquad (16)$$

$$\sum_{k\in\mathcal{K}_i} w_{ik} = 1 \qquad \forall i\in\mathcal{I} \qquad (17)$$

$$\delta^+_{ch}, \delta^-_{ch} \in \mathbb{N}^+ \qquad \forall c\in\mathcal{C}, h\in\mathcal{H}$$

$$w^I_{ik} \in \{0,1\} \qquad \forall i\in\mathcal{I}, k\in K$$

IP1

The objective is to minimize the total difference between the inbound pallet supply and the outbound pallet demand, while respecting the wishes regarding the inbound truck time windows. Constraint set (15) defines δ^+ and δ^- as described above. Constraint set (16) ensures that the number of inbound doors is enforced, while constraint set (17) makes sure that only one time window is assigned to each inbound truck.

In the second step noted IP^{*1}, the output of IP1 w^I_{ik} is used as a data to run IP^{*1}. IP^{*1} is similar to IP*, except for the fact that w^I_{ik} is no longer a decision variable but rather an input data. Constraint sets (4) and (12) are therefore discarded in IP*[1] – see Appendix C. Note that the term of the objective function which includes w^I_{ik} is not removed, although it is now a constant, so that the objective value stays comparable to the results of IP*.

2.3.2 *Heuristic 2*

IP2, the first step of heuristic 2, builds a feasible outbound truck schedule independent of the inbound data. The objective is to minimize the earliness and tardiness of the outbound trucks. Then, considering the outbound data fixed, IP*[2] is used to generate the inbound truck schedule.

Integer program IP2 uses w^O_{ok} as the only decision variables. It is formulated as follows:

$$
\begin{array}{lll r}
\min & \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} P^O_{ok} w^O_{ok} & & \\
& & & \\
\text{s.t.} & \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} W^O_{okh} w^O_{ok} \le N^O & \forall h \in \mathcal{H} & (18)\\
& \sum_{k\in\mathcal{K}_o} w_{ok} = 1 & \forall o \in \mathcal{O} & (19)\\
& w^O_{ok} \in \{0,1\} & \forall o \in \mathcal{O}, k \in K &
\end{array}
$$

IP2

The objective is to minimize the outbound transport providers' dissatisfaction. Constraint set (18) ensures that the number of trucks in use at any time period does not exceed the number of outbound doors, while constraint set (19) makes sure that only one time window is assigned to each outbound truck.

In the second step, the output of IP2, w^O_{ok}, is used as a data to run IP*[2] – see Appendix C. Similarly to what was done for heuristic 1, the formulation of IP*[2] is very similar to IP*, except for the fact that w^O_{ok} is no longer a decision variable but rather an input data. Constraint sets (5) and (12) are thus discarded in IP*[2]. The term of the objective function which includes w^O_{ok} is not removed, for the sake of comparison.

2.3.3 *Heuristic 3*

Heuristic 3 aims at finding a good truck schedule through a tabu search. Each solution is characterized by its truck schedule only: the detailed pallet moves are obtained from the truck schedule using three different methods based on solving a maximum flow problem.

The main elements of the tabu search are as follows; the complete algorithm can be found in algorithm 2.1.

```
sBest = initialSolution
sBestValue = Π0(sBest)
tabuList = new List
nbIterations = 0
while nbIterations<nbNonImprovingIterations do
    sValue = +∞
    for each sCandidate in the neighborhood do
        sCandidateValue = α0Π0^α(sCandidate)+ β0Π0^β(sCandidate)
        if sCandidate∉tabuList & sCandidateValue<sValue then
            s = sCandidate
            sValue = sCandidateValue
        end
    end
    tabuList.add(s)
    if tabuList.size > maxTabuListSize then
        tabuList.removeLast
    end
    nbIterations++
    if Π0(s) < Π0(sBest) then
        sBest = s
        nbIterations = 1
    end
end
return sBest
```

Algorithm 2.1: Tabu search algorithm for H3

TABU LIST. The maximum size of the tabu list (maxTabuListSize) is set to 7 as suggested by Glover [80].

STOPPING CRITERIA. The tabu search stops if the objective value does not improve after a fixed number of iterations (noted nbNonImprovingIterations in algorithm 2.1). We set this value to 5000.

INITIAL SOLUTION. To find an initial solution, IP* is run for a short amount of time, *e.g.* 2 seconds. The search is stopped before optimal. The solution obtained may or may not be feasible at this stage. For big instances, IP* might not be able to obtain a feasible solution within the time limit. The time limit can then be increased (for instance up to 240 seconds to solve all the instances in set12+12). For larger instances (set25+25), the result of heuristics H1 or H2 can be chosen as initial solution. The tabu search is used in this case to improve the result obtained by the heuristics.

NEIGHBORHOOD. Recall from section 2.1.3 that the sets $\mathcal{K}_i$ (resp. $\mathcal{K}_o$) of possible presence slots are completely enumerated for all inbound (resp. outbound) trucks. The enumeration is made by ascending starting date and ascending length, thus all the possible slots $\mathcal{K}_i$

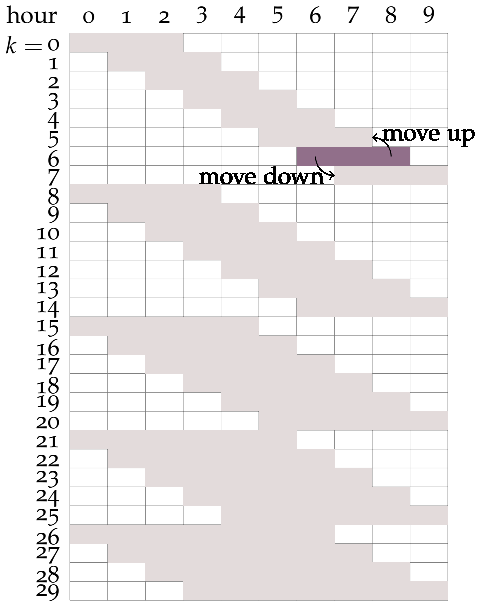


Figure 2.10: Neighborhood of a given slot

are indexed in a logical order (see Figure 2.10). For a given solution of the truck scheduling problem, a neighbor is obtained by changing its allocated slot k with the slot indexed by $k+1$ (move up) or with the slot indexed by $k-1$ (move down). A move up from the slot of index $|\mathcal{K}|-1$ gives the slot of index 0, and a move down from index 0 gives the slot of index $|\mathcal{K}|-1$. Most of the moves just shift the current slot one hour earlier or one hour later. Other moves make bigger changes (*e.g.* moving from $k=7$ to $k=8$ in Figure 2.10) and enable diversification during the search. Note that some solutions generated in this manner can be unfeasible. Unfeasible solutions are not excluded from the search because they can lead to better feasible solutions.

The complete neighborhood of a given solution can therefore be obtained by moving up and down all the inbound trucks and all the outbound trucks in the solution, which generates $2 \times (|\mathcal{I}| + |\mathcal{O}|)$ different neighbors. The algorithm selects the "best" of those neighbors which is not already in the tabu list. The choice is based only on the value of $\alpha_0\Pi_0^\alpha + \beta_0\Pi_0^\beta$ because the value of Π_0^γ is computationally expensive to evaluate (see below).

OBJECTIVE EVALUATION. The objective Π_0 is obtained with the same formula used in IP*: $\Pi_0 = \alpha_0\Pi_0^\alpha + \beta_0\Pi_0^\beta + \gamma_0\Pi_0^\gamma$. Penalties regarding the inbound trucks (Π_0^α) and the outbound trucks (Π_0^β) are obtained directly from the truck schedule characterizing the solution and from the penalty matrices P^I and P^O. In order to calculate Π_0^γ, it is necessary to know how exactly the pallets are transferred from and to the different trucks present. Three different methods are used to deduce this information from the truck schedules: they are detailed in the three sections that follow.

2.3.3.1 *Relaxed integer program*

The first option uses the same idea already used in heuristics 1 and 2. In order to find the optimal flow of pallets when the truck schedules are fixed, a version of IP* is run with w^I and w^O being fixed. The resulting IP model is noted IP*3 – see Appendix C.

2.3.3.2 *Maximum flow graph*

The goal is to find the value of Π_0^γ, *i.e.* to determine how many pallets go to storage when transferred in the best possible way from inbound to outbound trucks for which the time window of presence is fixed.

Another way to look at the problem is to determine how many pallets are transferred directly, without going through storage. To achieve this, the transfer problem can be modeled as a single-sink, single source time-expanded flow network. The maximum flow in the network is the maximum number of pallets that can be transferred directly from inbound trucks to outbound trucks. It is then assumed that the remaining pallets go to storage, which gives Π_0^γ.

Figure 2.11 shows how a given instance is transformed into the corresponding flow network. Each column in the graph represents a time interval h. At each time interval, the trucks present are modeled by a set of vertices of different colors, representing different clients. A single source provides each truck i, on its arrival date, with the right amount of pallets of each color c (capacity Q_{ic} on the edge). Another

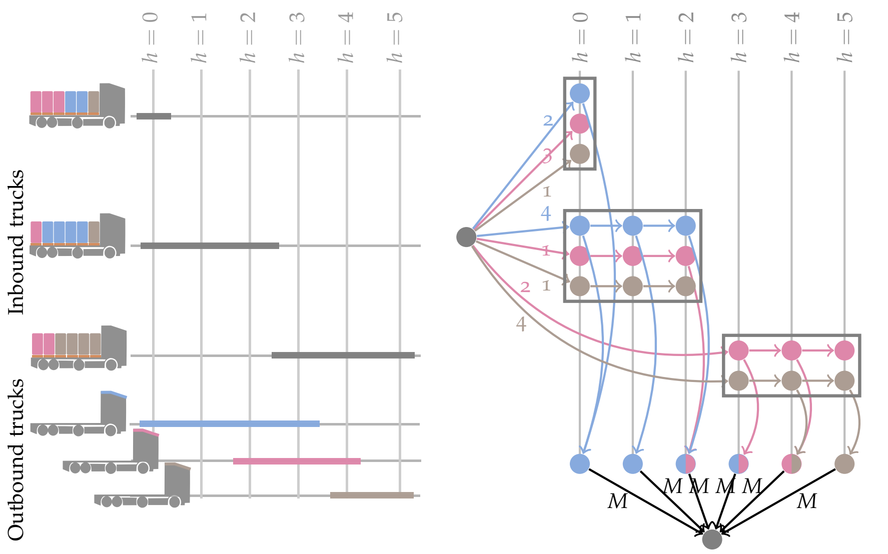


Figure 2.11: Transformation of an instance with fixed time windows into a maximum flow network

set of vertices, one vertex per time interval, represents the outbound side. If an outbound truck for client c is present on time interval h, then there is an edge between all the inbound truck vertices of color c at time h, and the outbound vertex at time h. All the outbound vertices are linked to a single sink by an edge of capacity M.

This graph model has the advantage of solving the transfer problem in polynomial time, for instance using a shortest augmenting path algorithm (see Ahuja *et al.* [5]). However it is not exactly equivalent to the relaxed Integer Program IP*3. The limitations are of two types:

1. The number of pallets put in each outbound truck is not limited in the graph model.
2. The quantity transferred from inbound trucks to storage cannot be included in the total transfer capacity.

For these reasons, the value of Π_0^γ given by the maximum flow algorithm can be slightly different from the optimal value given by IP* or IP*3. This difference is tolerable because the algorithm is used in a metaheuristic that does not guarantee optimality.

2.3.3.3 *Maximum flow integer program*

On a single run, solving the maximum flow problem described in the previous section is likely to be faster than solving an IP model. However, the value of Π_0^γ is solved every time a better solution is found in the tabu search, and only the capacities of a few edges change between two iterations. It is possible to exploit this property using CPLEX's ability to solve models in an iterative way: if the model has not been changed much in between, then CPLEX uses the previously found solution to find the new one, and behaves incrementally regarding changes of the bounds. In order to use this capability of CPLEX, the previous problem of finding a maximum flow in a graph is thus formulated as a Linear Program (LP).

Let us denote by $\mathcal{E}$ the set of edges and $\mathcal{V}$ the set of vertices in the graph, $\mathcal{A}(v)$ the set of edges entering vertex $v \in \mathcal{V}$, and $\mathcal{D}(v)$ the set of edges exiting vertex $v \in \mathcal{V}$. The decision variables are the flows

$$
\begin{array}{llll}
\max & \Delta & & \\
\text{s.t.} & f_e \leq C_e & \forall e \in \mathcal{E} & (20) \\
& \sum_{e\in\mathcal{A}(v)} f_e = \sum_{e\in\mathcal{D}(v)} f_e & \forall v \in \mathcal{V} \setminus \{s,t\} & (21) \\
& \sum_{e\in\mathcal{D}(s)} f_e = \Delta & & (22) \\
& \sum_{e\in\mathcal{A}(t)} f_e = \Delta & & (23) \\
& f_e \geq 0 & \forall e \in \mathcal{E} &
\end{array}
$$

IP$^{\text{max flow}}$

f_e, constrained by capacities noted C_e on every edge $e \in \mathcal{E}$. $\text{IP}^{\text{max flow}}$ is then written as a classical maximum flow model (see the LP model on the preceding page).

2.3.4 *Numerical experiments on the heuristics*

In this section, the heuristics described above are tested in order to assess their performances regarding computation time, compare their results to the optimal solution when possible, and see in which situation each heuristic provides good results.

2.3.4.1 *Comparing different versions of* H3

In this section, the different versions of H3 detailed in section 2.3.3 are tested and compared.

For small instances such as `set3+3`, the initial solution is likely to be optimal already, thus the tabu search afterwards is useless. For big instances on the other hand, it is possible that no initial solution is found within the time limit: it is the case on all ten instances of `set25+25`.

Figure 2.12 shows the results of the tests on the set of medium size instances (`set12+12`). On average on all ten instances, the fastest option is the method using a maximum flow integer program; it is also the method that yields the smallest objective function result (although it is really close to the relaxed integer program in this matter). When using heuristic H3 in the rest of the document, we will therefore use the maximum flow integer program to evaluate the objective function.

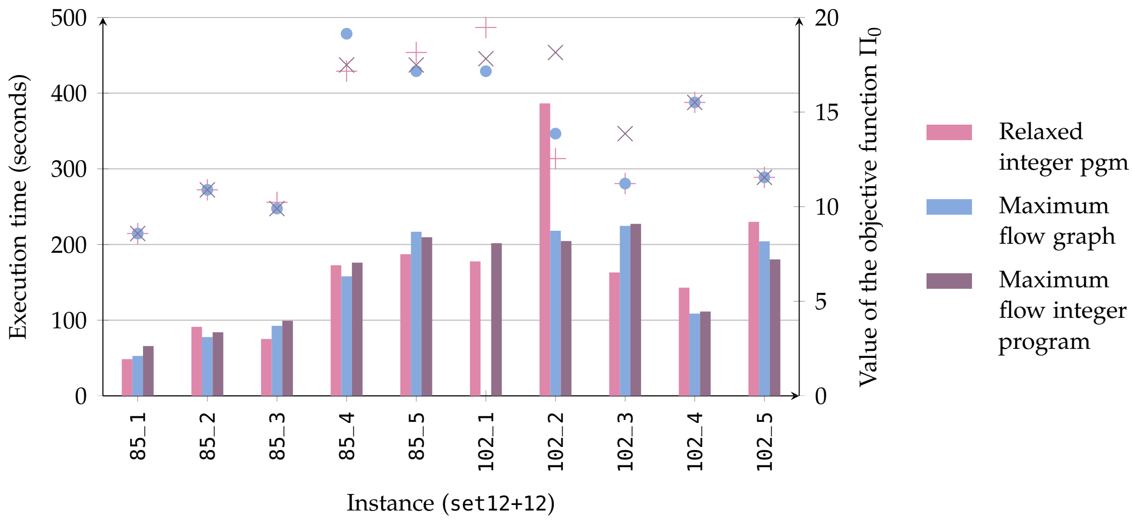
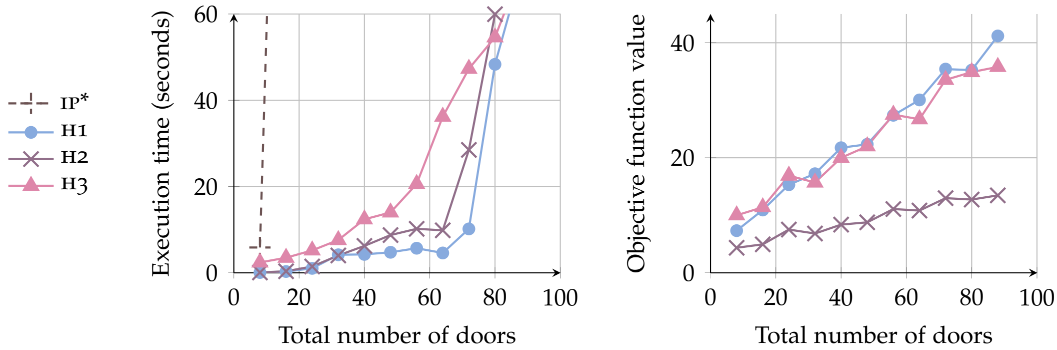


Figure 2.12: Execution time and value of the objective function for the three versions of H3

2.3.4.2 *Comparing the three heuristics*

In section 2.2.4 it is shown that IP* can only be used for very small instances due to an unreasonable execution time. In order to compare this result with those of the heuristics, the experiment settings used are similar to section 2.2.4. The instance generator is used with parameters $|\mathcal{H}| = 10$, $|\mathcal{D}| = 4$, $M = 4$ and $F = 4$. Setting the concentration of trucks equal to 0.4 truck/door/hour, the total execution time of the heuristic is monitored when the number of doors (inbound + outbound) increases. Coefficients α_0, β_0 and γ_0 are all equal and set to 0.33. Each value in Figure 2.13 is the average of the execution times or objective value obtained for 10 different instances, generated randomly from the parameters, as explained in section 2.2.3. Figure 2.13 displays the results regarding the execution time on the left hand side, and the value of the objective function on the right hand side.



The concentration ratio is kept fixed: $R = 0.4$

Figure 2.13: Results on IP*, H1, H2 and H3 as a function of the number of doors

First of all, we note that the heuristics are 75 times faster than IP* on the average for 2 doors and 8 trucks in total. The execution of IP* for 4 doors and 16 trucks takes 205 seconds on average it is not represented in the figure to avoid stretching the scale too much. H1, H2 and H3 are about 570 times faster than IP* in this case.

H1 can be computed in less than 10 seconds with up to 72 doors in the platform, whereas H2 can only handle 64 doors in 10 seconds. Within one minute, we can get a result for 80 doors. We note that the execution time increases exponentially beyond 80 doors. H3 is slower that H1 and H2, but its execution time does not increase as fast for bigger instances.

Regarding the quality of the result obtained, H1 and H3 are rather equivalent and clearly dominated by H2, which gives a result 40% smaller on average.

The results on the instance sets introduced in section 2.2.3 are displayed in Table 2.5 for `set12+12` and Table 2.6 for `set25+25`.

They show the same general tendency as described in Figure 2.13 for smaller instances. Since the procedure described in section 2.3.3

cannot find an initial solution for `set25+25`, the results for H3 in Table 2.6 are obtained using H2 as an initial solution. The tabu search enables to find a better solution for only one instance (`272_3`) in that case.

	H1		H2		H3	
	Exec time (s)	*Obj value*	*Exec time (s)*	*Obj value*	*Exec time (s)*	*Obj value*
`85_1`	1.5	23.76	17	4.62	65.8	8.58
`85_2`	1.3	28.71	35	8.58	84.1	10.89
`85_3`	1.6	24.42	40	10.56	99.2	9.90
`85_4`	1.5	34.65	59	17.16	176.0	17.49
`85_5`	1.6	42.57	560	23.43	209.6	17.49
`102_1`	1.8	21.45	4	13.20	201.7	17.82
`102_2`	2.6	16.50	101	10.89	204.6	18.15
`102_3`	1.9	25.08	61	7.92	227.2	13.86
`102_4`	1.4	27.72	34	15.18	111.4	15.51
`102_5`	2.1	32.01	118	12.87	180.3	11.55

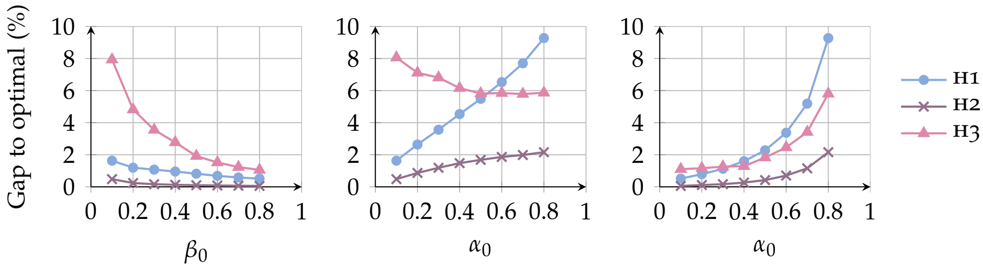
Table 2.5: Results of H1, H2, H3 on instance `set12+12`

	H1		H2		H3	
	Exec time (s)	*Obj value*	*Exec time (s)*	*Obj value*	*Exec time*	*Obj value*
`255_1`	10.9	41.91	13	16.83	136	16.83
`255_2`	9.6	44.55	121	18.81	143	18.81
`255_3`	11.7	36.63	24	13.20	209	13.20
`255_4`	86.0	50.16	out of memory		no initial solution	
`255_5`	76.3	58.74	9452	29.40	242	29.40
`272_1`	84.2	54.78	229	31.35	142	31.35
`272_2`	311.4	42.9	338	22.11	153	22.11
`272_3`	50.8	47.52	57	24.75	209	13.33
`272_4`	92.1	46.53	out of memory		no initial solution	
`272_5`	69.9	44.22	8567	36.30	193	36.30

Table 2.6: Results of H1, H2, H3 on instance `set25+25`

2.3.4.3 *Sensitivity analysis for the heuristics*

In the second set of tests, the performance of heuristics H1, H2 and H3 is compared to IP*. The number of doors ($N^I = N^O = 4$) and the number of trucks ($|\mathcal{I}| = |\mathcal{O}| = 8$) are therefore fixed. They are deliberately small (concentration 0.2 trucks/door/hour) so that the computation time of IP* stays reasonable. For each dot on Figure 2.14, 10 different instances are generated from the data parameters. The figure displays the average difference between the objective values of the heuristics and the optimal value given by IP*. Coefficients α_0, β_0 and γ_0 vary such that $\alpha_0 + \beta_0 + \gamma_0 = 1$.



(a) $\alpha_0 = 0.1$ (b) $\beta_0 = 0.1$ (c) $\gamma_0 = 0.1$

Figure 2.14: Distance to optimal when using heuristics

We observe that H2 always performs better than H1 and H3. However, when α_0 is small, the results are very close to the optimum for both (H1 and H2 (less than 5% of deviation for H1 and 2% for H2). It means that both heuristics perform well when the inbound truck schedule penalties do not weight much in the objective function. The best performance of H3 are for small values of γ_0, and H3 is better than H1 when both β_0 and γ_0 are small.

The results of H1 and H2 are almost insensitive to changes in β_0, the parameter weighting the outbound truck schedule penalties. It is the consequence of the fact that both heuristics focus primarily on the performance of the outbound truck schedule: H2 fixes the outbound schedule, while H1 fixes the inbound schedule subject to the synchronization of the inbound and outbound plans. H3 puts the same weight on the inbound and outbound truck schedules, but does not evaluate the value of Π_0^γ at each iteration, which explains the deterioration of its result when γ_0 increases.

Use H2 if the instance is not too big. For big instances, prefer H1 if α_0 is small and H3 if γ_0 is small.

H2 is less sensitive than H1 and H3 to the changes in parameters α_0, β_0 and γ_0, and performs quite well compared to IP*: Figure 2.14 shows less than 3% of deviation for any combination of α_0, β_0 and γ_0. Therefore, H2 can be used to solve any instance of reasonable size. However, for big instances with small α_0, H1 may be more interesting to use since its execution time is shorter, and the results do not deteriorate much. H3 will be preferred for big instances with small γ_0, and can also be used to improve the solutions found by the other two heuristics.

2.4 CONCLUSION

This chapter studies a truck scheduling problem with time windows for the inbound and outbound trucks, minimizing the quantity stored and the dissatisfaction regarding the time windows allocation. The problem is shown to be NP-hard in the strong sense.

Three heuristics are proposed in order to shorten the time needed to obtain a satisfying solution. The first two heuristics use a decomposition into two sub-problems, modeled by IP models used sequentially. The third heuristic is a tabu search in which the evaluation of a solution is done via an IP model or a network flow problem. Numerical experiments show that those three formulations can solve bigger problems, even if they cannot scale up to the biggest platforms with 150 inbound and 150 outbound doors.

Possible perspectives for this work would be to formulate, study and compare other heuristics, especially heuristics using a rolling horizon. The results of H1 or H2 could also be used as starting points to run IP*, which could significantly improve its execution time. H3 could be improved by implementing the maximum flow algorithm in an iterative manner, similarly to what was done for the maximum flow IP model. Other meta-heuristics used in the literature (see Table 1.6) could also be explored.

Two elements identified in section 1.2.5 as important gaps between theory and practice have not been addressed in this chapter, but could be added in the integer programs without too much difficulty: a mixed service mode (where doors can be used as inbound or outbound doors as needed) and a limited storage capacity. The effects of such modifications on the different models, their results and performances, should be investigated.

The models developed in this chapter are deterministic. What happens if the input data are not totally reliable – for instance if the transfer time is not constant, or if a truck arrives later than planned? Is the schedule able to absorb these variations without too many perturbations? Chapter 3 aims at answering these questions.

Mais il y a pire que l'imprévu,
ce sont les certitudes !

— Daniel Pennac

Chapter 3

ROBUSTNESS EVALUATION WITH SIMULATION

The models proposed in chapter 2 give optimal or close to optimal schedules in a deterministic case. However, the actual realization of the schedule is subject to uncertainties. How is the initial schedule perturbed in case of unplanned events, for instance if a truck is late or early? A discrete-event simulation is developed with FlexSim© to answer this question. After validation and verification, the simulation model is used to evaluate the optimization model subject to different sources of variations – on the time needed to transfer or unload a pallet, and on the actual arrival time of the trucks in the platform.

Based on numerical experiments, robustness metrics are proposed to evaluate the robustness of the schedule.

The work presented in this chapter is also presented in the following articles:

Ladier, A.-L., Greenwood, A. G., Alpan, G., and Hales, H. 2014. Issues in the complementary use of simulation and optimization modeling. *Les Cahiers Leibniz 211*.

Ladier, A.-L., Alpan, G., and Greenwood, A. G. 2014. Robustness evaluation of an IP-based cross-docking schedule using discrete-event simulation. In *Industrial and Systems Engineering Research Conference*. Montréal, Canada.

EVALUATION DE ROBUSTESSE PAR LA SIMULATION

Les modèles décrits dans le chapitre 2 fournissent une planification optimale, ou proche de l'optimal, dans un cas déterministe. La situation réelle est plus incertaine. Comment le planning initial est-il perturbé en cas d'événements imprévus, par exemple si un camion arrive en retard ou en avance ?

Pour répondre à cette question, nous avons développé un modèle de simulation à événements discrets avec le logiciel FlexSim©. Ce modèle reproduit le fonctionnement d'une plateforme de cross-docking, et utilise comme donnée d'entrée le planning de camions obtenu grâce aux modèles du chapitre 2. Le détail des mouvements de palettes n'est, en revanche, pas utilisé comme donnée d'entrée, pour que le modèle puisse s'adapter en cas de changement par rapport au planning prévu. Un algorithme simple est donc proposé pour organiser le flot de palettes.

Afin de vérifier et valider le modèle de simulation, il est nécessaire de s'assurer que dans un cas déterministe, le modèle de simulation se comporte de façon similaire au programme linéaire. Une analyse des différentes causes de déviation et des moyens de contourner ces problèmes est proposée.

Le modèle de simulation est ensuite utilisé pour évaluer la robustesse du modèle d'optimisation face à des perturbations à trois niveaux :

– variations sur la durée de transfert d'une palette au sein de la plateforme ;
– variations sur le temps de déchargement d'une palette ;
– variations sur les heures d'arrivée des camions à la plateforme (avance ou retard).

Le comportement du système est analysé en observant le nombre de palettes mises en stock, la déviation sur l'heure de mise à quai et la déviation sur le temps passé à quai, pour établir un lien entre le niveau de variabilité appliqué et les perturbations observées. A partir des résultats numériques, nous proposons trois indicateurs de robustesse, permettant d'évaluer numériquement la robustesse du modèle dans chacune des trois situations.

3 ROBUSTNESS EVALUATION WITH SIMULATION

In their 2010 review of crossdock truck scheduling problems, which also includes a research agenda listing the main issues left to be addressed in this area, Boysen and Fliedner note the following:

> "Arrival times of trucks are typically bound to heavy inaccuracies, because traffic congestion or engine failures delay inbound trucks [. . .]. Thus, the following research questions need to be answered in this context: up to which "level of uncertainty" are expected arrival times of trucks useful information to be considered in truck scheduling? How to derive robust plans, *i. e.* plans which remain feasible in spite of (shorter) delays?"
>
> Boysen and Fliedner [31]

This chapter addresses the first research question: up to which level of uncertainty do the models of chapter 2 hold?

As noted by Rohrer [167], simulation can be used to test different control algorithms before their implementation since it provides an environment that is rather close to real-life situations. Simulation is therefore proposed here as a tool to measure the robustness of the IP model and heuristics that were presented in chapter 2.

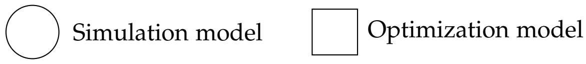
Section 3.1 shows how to create a simulation model that enables to evaluate the performances of an optimization model. Section 3.2 describes a robustness evaluation methodology, numerically tested in section 3.3 in order to propose in section 3.4 robustness metrics adapted to our problem.

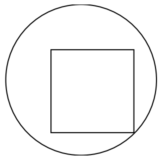
3.1 LINKING OPTIMIZATION AND SIMULATION

Optimization models and simulation models are usually built for different purposes and using different modeling rules. Combining them can provide complementary insights to a given problem but can also prove difficult because of their differences. This section shows how to build a simulation model that can be used to evaluate the optimization models described in chapter 2.

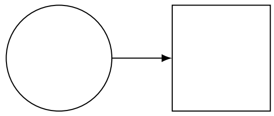
3.1.1 *Optimization-simulation in the literature*

Among the papers that combine optimization models with simulation in the cross-docking literature, we identify four different ways of combining simulation and optimization models. These relationships are illustrated in Figure 3.1 and described below.

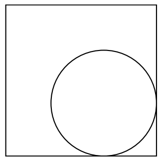




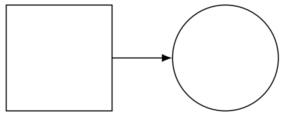
(a) Optimization within a simulation model



(b) Simulation model provides data to the optimization model



(c) Simulation within an optimization model



(d) Simulation evaluates optimization solutions under different conditions (usually stochastic environment)

Figure 3.1: Complementary uses of simulation and optimization models

(a) *An optimization model is embedded within a simulation model.* Wang and Regan [212] apply this technique with two time-based algorithms to schedule inbound trucks in a crossdock in real-time, with the aim of minimizing pallet transfer time. The algorithms are embedded in an *Arena* simulation model. In a different example, Clausen *et al.* [48] simulate the operations within a network of LTL terminals, using optimization (multi-stage mixed-integer program, solved with a modified tabu search) to make decisions regarding the routing between the different terminals.

(b) *The output of a simulation is used as input to an optimization model.* Hauser [100] in her dissertation uses a simulation (developed with *Arena*) of a Toyota manufacturing plant to test different crossdock layouts. The objective is to minimize the walking distance during the dispatching operation, with the idea of eventually balancing the workload. Genetic algorithms are used to decide where each destination goes in the best layout determined by the simulation. Another example is given by Liu and Takakuwa [133], who use a simulation model developed in *Arena* to determine the workload at a cross-docking center. Data from the simulation are then used as input in an IP model that produces an optimal schedule for the operators.

(c) *A simulation model is embedded within an optimization model.* This method, often called simulation-optimization, is widely used in diverse fields. In the cross-docking literature, McWilliams [140] generates an inbound truck schedule using this technique. A simulation model is used to evaluate the objective function after each permutation of the meta-heuristics. In a similar way, Aickelin and Adewunmi [6] solve the crossdock truck-to-door assignment problem with a local search (memetic algorithm); a simulation model evaluates the objective function at every iteration. Instead of using the simulation as a black box, Almeder *et al.* [7] translate the solution of the optimization model into decision rules for the discrete-event simulation, and apply the procedure iteratively until a stable point is reached.

(d) *The output of an optimization model is used as input to a simulation model.* Gambardella *et al.* [75] are, to the best of our knowledge, the only ones applying this technique in the logistics platform environment. They develop a discrete-event simulation model of an intermodal container terminal in order to check the validity of a resource allocation within the terminal, that is generated with an integer linear program. This work, carried out in 1998, relies on a custom-coded simulation program lacking the numerous functions of modern simulation software programs.

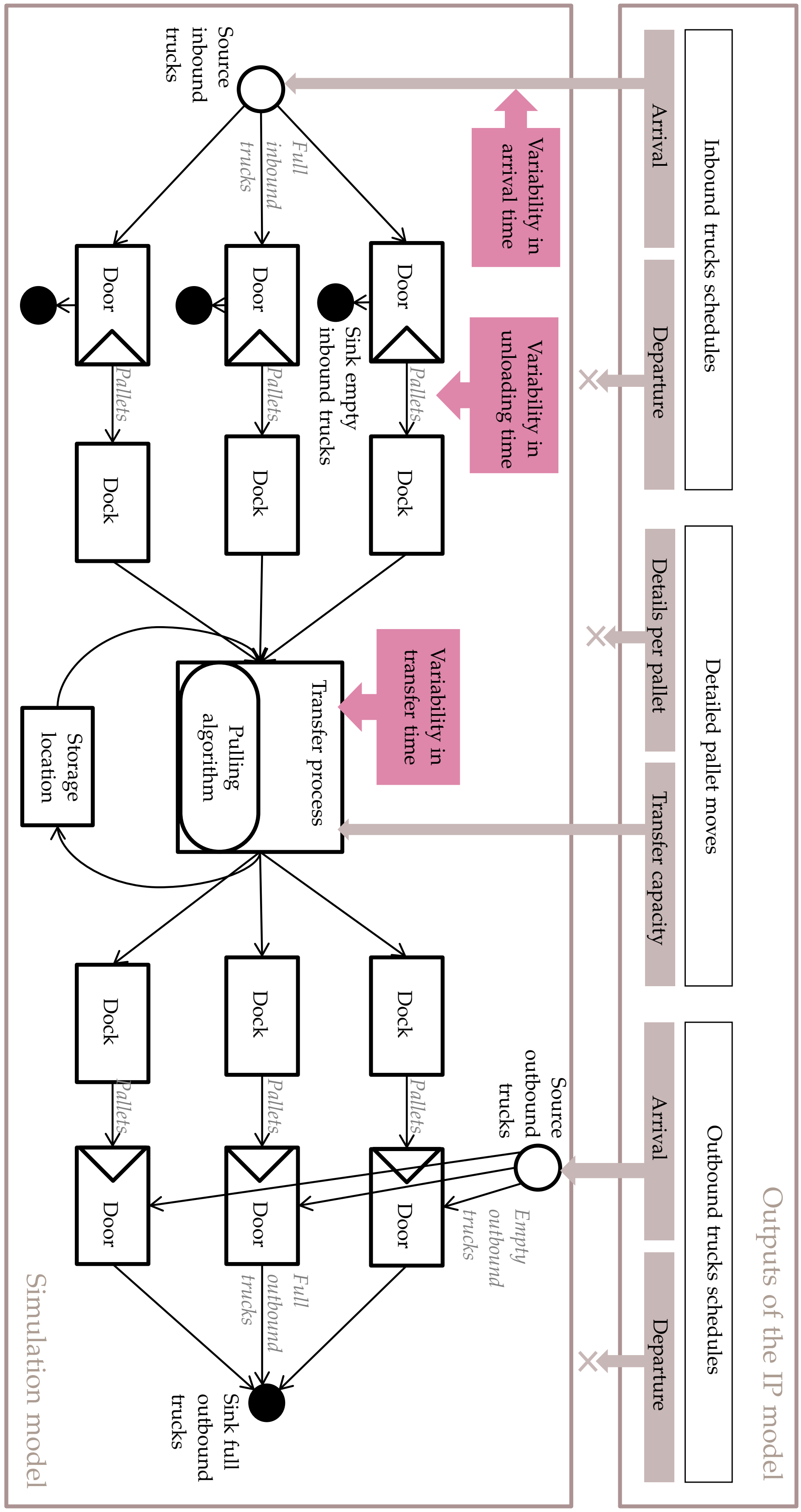
Our goal in this chapter is not only to fill the gap left in case (d), but also to evaluate the robustness of our previous models. In the case of cross-docking operations, we demonstrate the use of a simulation model to evaluate the robustness of a solution provided by an IP model.

3.1.2 *Model description*

The relationships between the simulation and optimization models are shown in Figure 3.2. As detailed in sections 2.2 and 2.3, the outputs of IP* and the heuristics are the truck schedules (arrival and departure times for the inbound and the outbound trucks) and the detailed pallet moves (number of pallets moved from one point to another at each time period).

The simulation model takes as input the truck arrival times that are determined by the IP model or one of the three heuristics depending on the instance size. It is assumed that the manager has called the transportation providers to set up their arrival time according to the optimization results. However, the truck departure time is not forced according to the optimization results: the inbound trucks leave when they are empty, and the outbound trucks leave when they are full.

For the simulation model to be able to react to a planning change, the pallet moves have to stay flexible. If each pallet was required to move only at the time and to the location decided by the IP model, the simulation would be totally blocked when a truck is late, or operators would stay idle in front of an early truck. Therefore, instead of using the exact times determined by the IP model to move each pallet, the simulation uses a greedy algorithm (algorithm 3.1) to decide which



Example with 3 inbound and 3 outbound doors

Figure 3.2: Links between optimization and simulation

for all outbound docks which need pallets available in the inbound side *do*
  find the dock o whose truck has to leave first
  note c the matching client;
end
if no dock was found *then*
  for all clients of the pallets present on the inbound side *do*
    find client c whose truck leaves last
  end
  set o as the storage location
end
for all inbound docks which have pallets for client c *do*
  find dock i whose truck has to leave first
end
Pull a pallet of type c from dock i to dock o.

Algorithm 3.1: Pulling policy

pallet (for which client c) is pulled from which inbound dock i, and sent to which outbound dock o.

The data on the pallet moves determined by the IP model, when aggregated, give information on how many pallets are moved per hour, and therefore what staffing levels are needed for the transfer at each time period. Assuming that the manager has staffed the platform accordingly, the output of the IP model is used to limit the hourly capacity of pallet transfer in the simulation model.

Figure 3.2 features a simplified flow diagram of the simulation model shown in Figure 3.3. The simulation model is developed using *FlexSim*© .

www.flexsim.com

3.1.3 *Model validation and verification*

To *validate* a model is to determine whether or not it is a meaningful and accurate representation of the real system, and contains sufficient accuracy to meet its intended use. *Verification* is the process of determining whether a model is working as intended.

Validation is about building the right model.
Verification is about building the model right.

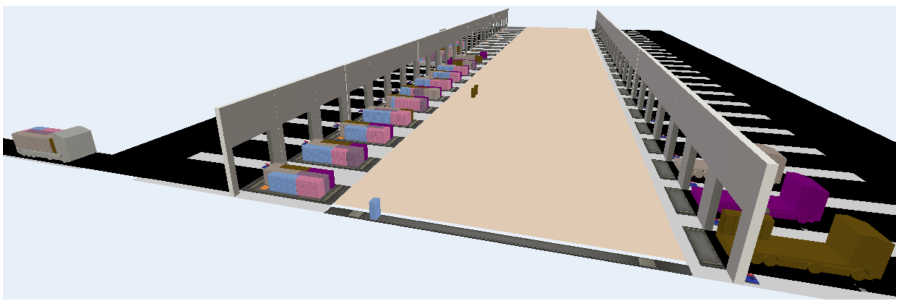


Figure 3.3: Screenshot of the simulation model

In order to validate and verify the simulation model, one expects it to behave similarly to the optimization model under deterministic conditions. The validation and verification of the simulation model is done in this section with the instances of `set3+3` described in section 2.2.3.

The next sections describe the disparities occurring between the two models (optimization and simulation) due to differences in the modeling approaches, and how these issues can be solved or circumvented for the validation of the model. Explanations on how validation is carried out can be found in section 3.1.3.6.

3.1.3.1 *Spatial representation*

The choice was made for the optimization model to ignore the spatial dimension (see section 2.1.1): the doors are interchangeable and the transfer time does not depend on the distance. Because of the spatial nature of the actions, such assumptions do not adapt well to discrete-event simulation. There is a trade-off to be made between fidelity to the optimization model and closeness to realistic operations. A compromise approach is thus adopted: the transfer time is controlled by making it a process step in the simulation instead of a distance- and speed-affected move from one point to another. The consistency of the simulation model with the optimization model is validated by setting the transfer time to zero.

3.1.3.2 *Transfer logic*

The logic implemented by the simulation model through algorithm 3.1 is close to what a manager would do; however, it does not give the optimal solution (*i. e.* exactly the same solution as the one given by the IP model) in all cases. In some cases, it leads to having outbound trucks leaving earlier than planned while inbound trucks leave late. The simulation can be driven towards a solution closer to the optimal, but it cannot determine the optimal solution unless it embeds an optimization module (this is case (a) in Figure 3.1, and beyond the scope of this work).

3.1.3.3 *Transfer rate and resources*

The optimization model only determines a given amount of tasks that have to be carried out within a given time interval: its output does not give information about the order, the batch size, the parallelism of the tasks. The simulation needs to have information (or to make choices with its default internal logic) on the resources that carry each task.

Let us assume that $M = 30$ pallets/hour: there are different ways of modeling such a transfer rate. The first option consists in using three resources at a rate of 10 pallets/hour each; the second option uses one resource at a rate of 30 pallets/hours. Those two different ways of modeling do not give the same results over a given time interval. If an outbound truck arrives at 10:00, then any pallet transferred from the inbound side before that time goes to storage, while any pallet processed after 10:00 goes directly into the outbound truck. In the first option, each pallet needs 6 minutes to be transferred. Therefore, between 9:55 and 10:00, no pallet is fully transferred and no pallet goes into storage. In the second option, the resource transfers each pallet in 2 minutes. Therefore, between 9:55 and 10:00, two pallets are processed and they both go into storage.

Because it is more realistic and creates less unnecessary storage, the first option (multi-channel process) is chosen. In the simulation model, we thus assume that one resource can process 17 pallets per hour (*i. e.* takes about 3.5 minutes to scan, transfer and load a pallet). The number of resources R is set such that $R = \frac{M}{17}$.

For detailed calculations of the standard times, see *Appendix D.*

3.1.3.4 *Transfer capacity*

By its nature, simulation is greedy, *i. e.* it processes as many pallets as possible in one event while the IP model can transfer less pallets per time period if it improves the objective function. In order to force the simulation model to obtain a result similar to the optimization model, it is thus necessary to limit the amount of pallets that can flow through the model during each time period. This is done by using the output of the IP model (number of pallets transferred per time interval) to determine the capacity of the transfer process in the simulation model. This capacity, *i. e.* the number of available resources in the multi-channel process modeling the transfer, vary through time.

3.1.3.5 *Time representation*

The granularity of both models is different: the optimization model uses discrete time intervals of *e. g.* half an hour or one hour, whereas in discrete-event simulation, events occur at precise instances of time, *e. g.* a truck arrives 27.1752 minutes after the arrival of the previous truck. IP* only allows a truck to leave at a multiple of 60 minutes, while the trucks in the simulation model leave at any time; they leave when a specified condition is met, *e. g.* when a truck is empty (inbound) or full (outbound). Therefore, the difference between the trucks departure time as calculated by the optimization model and the trucks departure time as observed in the simulation, can be as large as 59 minutes even though the system behaves as expected. Those gaps can be reduced by shortening the time intervals used in the optimization model; however, that makes the optimization model more complex (and possibly incomputable) and some gaps will always be observed. In order to circumvent this issue, performance is measured in terms of intervals, as detailed in the next section.

3.1.3.6 *Model validation*

In order to check that the simulation model is an accurate representation of the optimization model, it is run under deterministic settings, without adding any source of uncertainty. Using as input the schedules calculated by IP* or H2 for each instance of `set3+3`, `set12+12` and `set25+25`, we check that the schedule is correctly realized in this deterministic setting for each instance. This is done by ensuring that the values of Π_0^α, Π_0^β and Π_0^γ in the objective function of the optimization model are close to the experimental values measured in the simulation model.

3.2 EVALUATING THE ROBUSTNESS OF THE IP MODEL

The performance indicators needed when testing an optimization model with a simulation model differ from the indicators that would classically be used in a simulation. The main goal here is to compare the performance of the simulation model in the deterministic case with its performance when some elements of the model follow random distributions.

3.2.1 *Robustness evaluation using simulation in the literature*

The robustness literature gives several examples of robustness evaluation through simulation. Leon *et al.* [117] propose slack-based robustness measures and evaluate them with a simulation study. Valckenaers *et al.* [198] review simulation-based studies that analyze scheduling problems, especially rescheduling techniques (repairing the schedule after an unexpected event occurred). They propose a method to evaluate the different rescheduling techniques. In [202], van de Vonder *et al.* conduct a simulation experiment to investigate whether it is beneficial to concentrate safety time in *project buffers* (positioned at the end of the critical chain) and *feeding buffers* (inserted when a non-critical chain activity joins the critical chain), or whether it is preferable to insert time buffers that are scattered throughout the baseline project schedule in order to maximize schedule stability. They show how to choose the buffering strategy depending on the characteristics of the project. In another article, van de Vonder *et al.* [201] propose different algorithms to include time buffers in a project schedule, and evaluate these algorithms with a simulation-based analysis. Canon and Jeannot [39] compare different robustness metrics used in the literature by performing an experimental study and showing how the different metrics relate to each other. Hazır *et al.* [101] propose a number of performance measures for robust project scheduling. They use a Monte-Carlo simulation to see which of these measures have the highest correlation with indicators on the project punctuality.

Those different papers deal mostly with project scheduling. Our goal is to propose a method that is applicable to cross-docking operations.

3.2.2 *Robustness evaluation methodology*

This section details the methodology used to evaluate the robustness of truck schedules obtained with an IP-based model that inputs deterministic data.

3.2.2.1 *Modeling variability*

How disrupted is the system subject to stochastic events? To answer this question, three possible sources of variability are considered:

– Time needed to complete the *transfer* of a pallet due to the performance of the workers doing the transfer, the traveling distance, or the congestion of the platform.
– Time needed to *unload* a pallet due to the way trucks are loaded, number of workers working on the same truck, and skills of the workers. Both activities, transfer and unloading, are not explicitly taken into account in the IP model: they are assumed to be performed in masked time. Thus, it is interesting to see how sensitive the schedule is to variations in process times.
– *Truck arrival times* due to, for example, traffic congestion or bad weather conditions.

We next describe how variability regarding transfer and unloading time, as well as truck arrival time, is inserted into the simulation model.

TRANSFER AND UNLOADING TIME. Transfer and unloading times are modeled in the simulation with triangular distributions. Such distributions can be used when limited data about a process is available (Jannat and Greenwood [104]). It also has the advantage of being bounded, which is not the case of *e. g.* normal or exponential distributions. A triangular distribution is defined by its location parameters: a (minimum value), b (maximum value) and m (mode).

The minimum and maximum values of the unloading time and transfer time are detailed in the "Standard process time" column of Table 3.1 on the following page. Those values are determined using the classic crossdock sizes given by Bartholdi and Gue [19], and standard process times for logistic operations (Gauvreau [78]).

For detailed standard time calculations, see Appendix D.

The behavior of the system is to be tested when the variability of the transfer time increases; therefore, the coefficient of variation is increased while keeping a constant skewness (equal to 0 since the

		Standard process time	Experimental values			
			$c_v = 0.1$	$c_v = 0.2$	$c_v = 0.3$	$c_v = 0.4$
Unloading	a	3.5	3.10	2.09	1.09	0.08
	b	4.7	5.10	6.11	7.11	8.12
Transfer	a	2.8	2.67	1.80	0.94	0.07
	b	4.3	4.39	5.26	6.12	6.99

Table 3.1: Distributions parameters for unloading and transfer process times

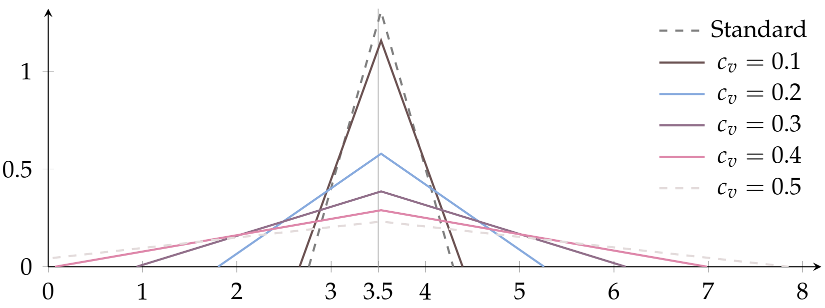


Figure 3.4: Transfer time triangular distribution

distribution is symmetric) and a constant mean. Using the table proposed by Jannat and Greenwood [104], we calculate the values of a and b when the coefficient of variation c_v increases. Since the distribution represents a process time, only the cases when $a > 0$ are kept. Only symmetric triangle distributions are used, for which the mode m equals the mean. This simplifying assumption is not contradictory with the industrial standard times (Appendix D), and using only symmetric distributions eliminates the bias a skewed distribution could introduce. The parameters of the resulting triangular distributions are shown in Figure 3.4 for the transfer time, and Table 3.1 for a synthesis of all values used. Note that the standard process times are closer to the case $c_v = 0.1$, which is thus the most realistic range for the transfer and unloading time.

TRUCK ARRIVAL TIME. The truck arrival times are defined by the IP model: to test the effect of variability in truck arrival times, what should be modeled in the simulation is only a deviation compared to this predefined arrival time. The deviations represent early or late arrivals; zero deviation means the corresponding truck arrives on time.

Let us call d the random deviation applied to each scheduled truck arrival time t_0 calculated by the IP model. Since most deviations are very short (a few minutes) and large deviations occur only occasionally, the arrival deviations d are assumed to be exponentially distributed, similarly to what is done by Wang and Regan [212]. Their mean is denoted by δ. In order to avoid unrealistically large time deviations, the distribution is truncated such that no value can be greater than $10 \times \delta$.

In order to determine whether the deviation corresponds to a late or an early arrival, a multiplier σ is defined such that

$$\begin{cases} \sigma = -1 & \text{for an early arrival} \\ \sigma = +1 & \text{for a late arrival} \\ \sigma = 0 & \text{for an on-time arrival} \end{cases}$$

The probability mass function for σ is specified as $P(\sigma = +1)$, $P(\sigma = -1)$ and $P(\sigma = 0)$. Therefore, for each truck arrival, its simulated arrival time is:

$$t_a = \max(0;\ t_0 + d \times \sigma)$$

where d and σ are random samples from their respective distributions.

3.2.2.2 *Measuring perturbations*

In order to measure the deviation between the performance of the realized schedule and the initial deterministic performance, the following measurement indicators are used:

TOTAL NUMBER OF PALLETS IN STOCK I_1

ERROR IN DOCKING TIME (inbound I_2 and outbound I_3): for each truck which docks later than expected, we compute the absolute difference between the scheduled docking time and the time at which the truck actually docks, in minutes. The indicator is the sum of those deviations for all inbound or outbound trucks.

ERROR IN STAYING TIME (inbound I_4 and outbound I_5): for each truck which stays docked longer than expected, we compute the absolute difference between the scheduled time spent at the dock, and the actual time spent at the dock by the truck, in minutes. The indicator is the sum of those deviations for all inbound or outbound trucks.

The time-related indicators I_2 to I_5 are only considered for the trucks that arrive and/or leave later than planned. Earliness is not explicitly taken into account in order to keep the number of indicators to follow reasonable. Part of the earliness situations do not impact the schedule: for example, a truck arriving early will have to wait if all doors are busy, and eventually be docked at the time originally planned for its arrival. If a door and the matching resources are available when the truck arrives, it can be unloaded early, which can impact the stock level. This side effect of early arrivals can be captured in I_1.

3.2.2.3 *Simulation parameters*

The simulation is run until the platform is empty and all trucks have left. This occurs after about 10 simulation hours due to the structure of the instances, but in some cases the operations are delayed and finish later.

The tests are run on instance `set3+3` and `set25+25` defined in section 2.2.3. The schedule for `set3+3` is obtained with the IP model. Since the size of the instances with 25 inbound and 25 outbound doors is too large to be handled by the integer program used previously, the truck schedules for `set25+25` are calculated using heuristic H2, which was shown to be faster and better for larger instances.

Each of the 21 instances of `set3+3` and `set25+25` is tested over a number of scenarios, *i. e.* a set of different values for the experiment parameters. Each scenario is tested over 20 different replications – with a confidence interval of 95%, this provides sufficient precision for analysis. For each replication, the value of each indicator I_i is compared to the value V_i of the deterministic case, checking whether or not this value is in the interval $V_i \pm \varepsilon_i$, where ε_i represents an acceptable tolerance for indicator i (ε_1 is a number of pallets, ε_2 to ε_5 are in minutes). The percentage of replications off-limits obtained depends on ε_i: see Figure 3.5. The values of the performance measures I_i are averaged for each scenario over the 20 replications and the 21 instances.

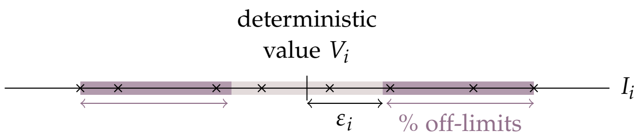


Figure 3.5: Definition of ε_i and percentage off-limits

Note that in general, the platform manager knows the tolerances ε of his/her organization. For example, some companies give financial penalties to their transportation providers if they are more than 15 minutes late; implicitly, the company assumes it can absorb delays smaller than 15 minutes, but not larger. In the next section, we use the simulation model to estimate ε in different cases, and propose robustness indicators linked to this tolerance.

3.3 NUMERICAL RESULTS

Following the methodology detailed in section 3.2.2, the simulation model is used to gather insights on the reaction of the IP schedules subject to variability (in transfer times, unloading times and truck arrival times). The objective is to propose a robustness indicator for each cause of variability studied.

3.3.1 *Variability in transfer time*

In this section, unloading time is equal to zero; transfer times are stochastic and follow a triangular distribution as detailed in section 3.2.2.1.

Figure 3.6 shows the value of the percentage off-limits for each indicator I_i, for different sets of values of ε_i, for instance `set3+3` and `set25+25`.

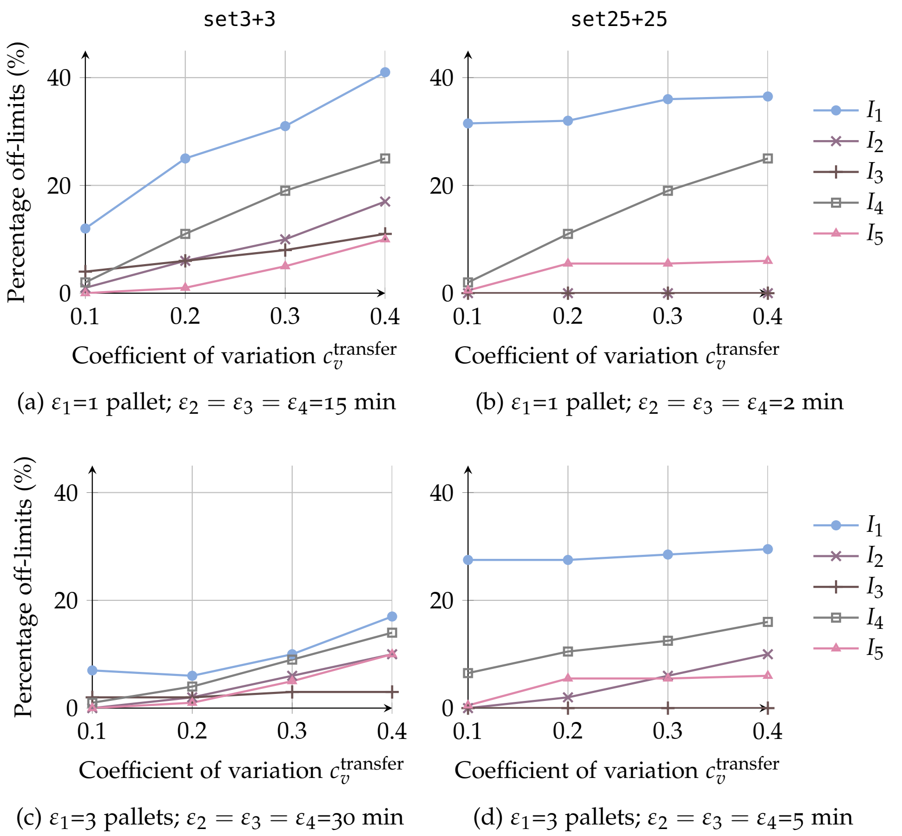


Figure 3.6: Percentage off-limits with different values of ε_i

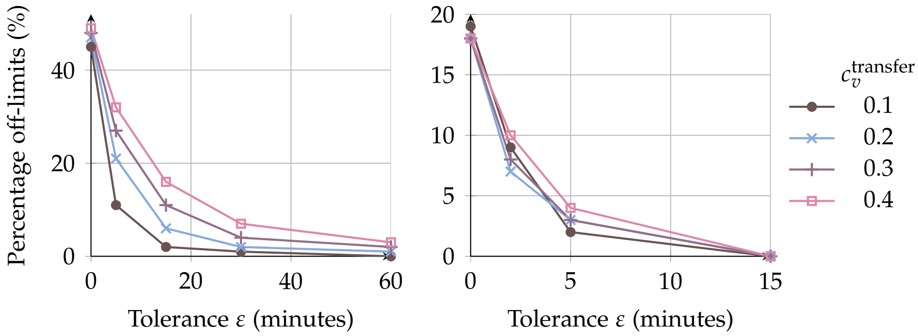
Figure 3.6 shows that a higher coefficient of variation leads to a higher percentage of results off-limits: the curves are monotonically increasing for all indicators, for both instance sets). The different indicators are less sensitive to changes in the coefficient of variation in the case of `set25+25`, compared to the case of `set3+3`. The total number of pallets in storage and the error in stay time for the inbound side are the most sensitive indicators when the transfer times become more variable. The outbound side indicators (I_3, I_5) are less sensitive for both instance sets.

Table 3.2 on the next page separates the indicators related to inbound and outbound trucks. We observe again that the outbound indicators are less sensitive. This is because the transfer algorithm favors the outbound side in the simulation.

		inbound (I_2 and I_4)		outbound (I_3 and I_5)	
		$c_v = 0.1$	$c_v = 0.4$	$c_v = 0.1$	$c_v = 0.4$
tolerance ε	0 min	65% / 27%	71% / 27%	26% / 10%	27% / 9%
	5 min	17% / 3%	44% / 8%	8% / 0%	21% / 0%
	10 min	4% / 0%	31% / 1%	2% / 0%	14% / 0%
	15 min	2% / 0%	23% / 0%	2% / 0%	12% / 0%

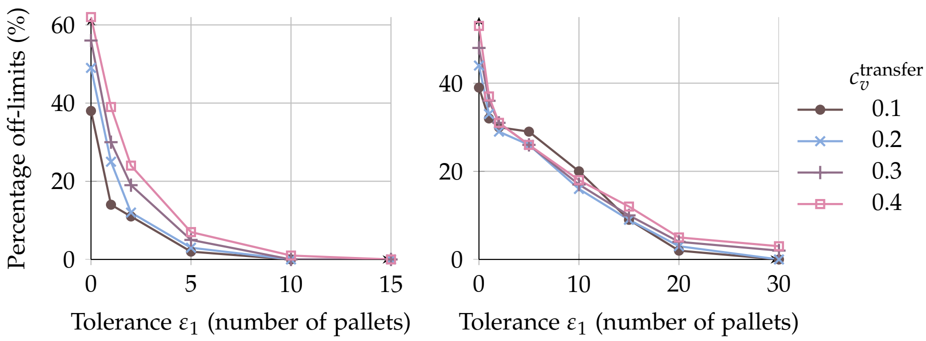
(set3+3 / set25+25)

Table 3.2: Transfer time variations: percentage off-limits for inbound- and outbound-related indicators



(a) set3+3 (b) set25+25

Figure 3.7: Transfer time variations: percentage off-limits for the average of I_2 to I_5



(a) set3+3 (b) set25+25

Figure 3.8: Transfer time variations: percentage off-limits for I_1

Nevertheless, since the shapes of the curves for the different indicators are similar, in the remaining of this chapter we aggregate the temporal indicators I_2 to I_5 together for the sake of readability.

Figure 3.7 shows how the average percentage off-limits (average of I_2 to I_5, with 20 replications for each instance) varies with different values of the tolerances, set such that $\varepsilon_2 = \varepsilon_3 = \varepsilon_4 = \varepsilon_5 = \varepsilon$.

For `set3+3` and for a coefficient of variation $c_v = 0.1$, the deviation drops to zero for $\varepsilon \geq 60$ minutes. The percentage off-limits is very sensitive for tolerances smaller than 15 minutes, and almost insensitive when the tolerances are greater than 30 minutes. Instances of `set25+25` are less sensitive than those of `set3+3`; this is due to their structure. Having a great number of doors provides more flexibility: when a pallet is unloaded it is more likely that a corresponding truck is available, even when the system is perturbed. A similar behavior is observed for indicator I_1: see Figure 3.8.

3.3.2 *Variability in unloading time*

In this set of experiments, the transfer time is deterministic and equal to 3.5 minutes. The unloading time is stochastic and follows the triangular distributions described in section 3.2.2.1. Results (Figure 3.9) show a pattern similar to the one in Figure 3.7: a higher coefficient of variation implies a higher percentage of cases off-limits, for all tolerance values.

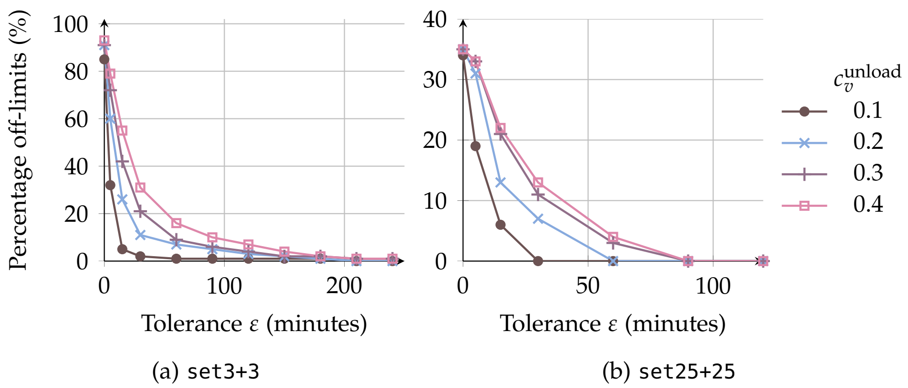
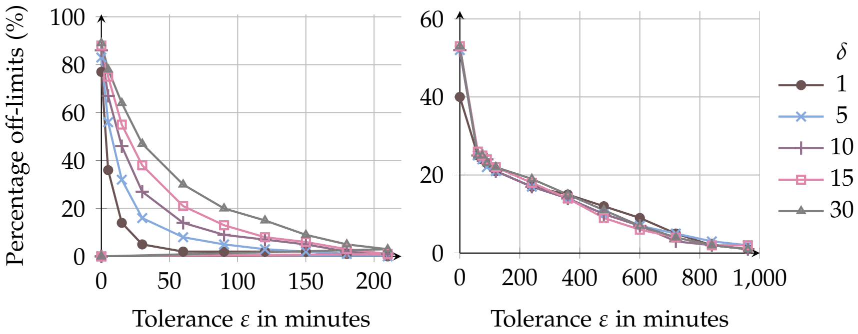


Figure 3.9: Unloading time variations: percentage off-limits for I_2 to I_5

3.3.3 *Variability in truck arrival time*

The percentages of trucks arriving late, early, and on time are varied such that the total is 100%. We observe the percentage off-limits (aggregated over I_2 to I_5) as a function of the tolerance ε, with different values of the mean delay δ. An example of result obtained when

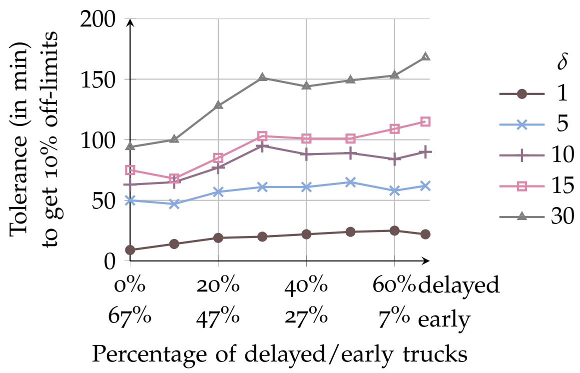


(a) set3+3 (b) set25+25

Figure 3.10: Truck arrival time variations: percentage off-limits for I_2 to I_5

40% of trucks arrive late, 27% are early, 33% unchanged) is displayed in Figure 3.10.

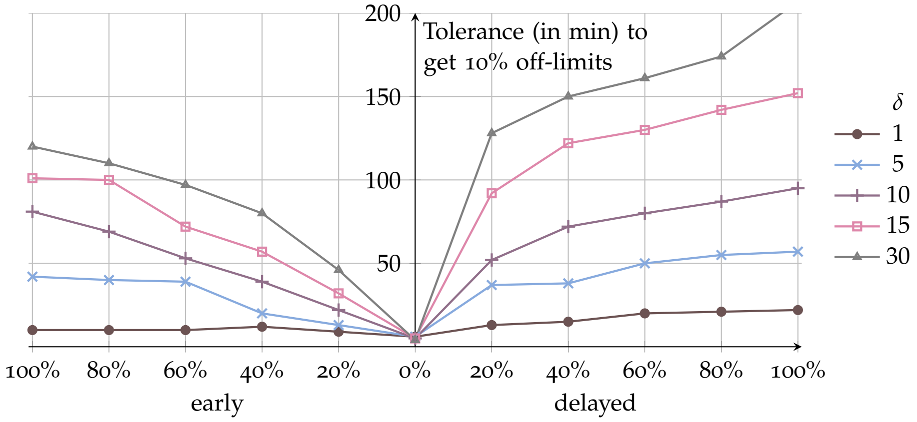
Again, the curves' patterns are similar to the observations made in sections 3.3.1 and 3.3.2. But the proportion of trucks arriving early, late and on time is a new parameter compared to what is done in the previous section. Let us suppose that 33% of the trucks arrive on time, and the remaining 67% are either delayed or early with varying proportions. Figure 3.11 shows the effect of this proportion, for different values of δ, on the tolerance to be set in order to get 10% off-limits.



set3+3 – Aggregated I_2 to I_5 – 33% of the trucks are on time

Figure 3.11: Tolerance to get 10% off-limits function of the truck punctuality

We note that the curves in Figure 3.11 are rather flat, which shows that early arrivals tend to compensate late ones. Tolerance ε is not very sensitive in that case: for example in set3+3, when the delays follow an exponential distribution of mean $\delta = 5$ min, the tolerance to get 10% off-limit is always around 50 minutes, no matter what the proportion of delayed/early trucks is.



set3+3 – Aggregating I_2 to I_5

Figure 3.12: Tolerance to get 10% off-limits function of the proportion of early/delayed trucks

In order to get more variations and thus a monotonic curve, we vary the percentages of trucks arriving late $P(\sigma = 1)$, or the percentage of trucks arriving early $P(\sigma = -1)$, but not both at the same time – the rest are trucks arriving on time. Figure 3.12 shows the tolerance to be set in that case, in order to get 10% off-limits for different values of δ. For example, if 20% of the trucks are delayed and the truck delays follow an exponential distribution of mean $\delta = 10$ minutes, the tolerance has to be set to 50 minutes.

Note also on Figure 3.12, that a given percentage of early trucks creates a smaller disruption than the same percentage of late trucks. This confirms the intuitive idea that delayed trucks are "worse" than early trucks – early trucks can wait, while the delay of a truck arriving late can be difficult to compensate.

3.3.4 *Correlation analysis*

While carrying out the numerical experiments, we noticed an interesting fact about the way the two indicators I_2 and I_4 (error in docking time and error in staying time in the inbound side) relate to each other. For instances for the set3+3, a linear correlation of coefficient $r \geq 0.75$ exists between these two indicators, as shown in Table 3.3 on the following page.

Recall that an error in docking time occurs when a truck cannot dock at the scheduled time, because the dock is occupied by another truck. This happens when the previous truck stayed docked for too long. When the linear regression line has a coefficient 1 (instances 17_3, 34_4 and 34_5), the total error in docking time is totally explained by the total error in stay time, *i. e.* a set of inbound trucks that stayed x minutes longer caused trucks that followed to dock x minutes late, exactly. We will refer to the trucks which stayed longer than

Instances	Coefficient of the regression line	Coefficient r
17_1, 17_2, 34_2, 34_3	0	-
17_4	0.15502	0.75
34_1	0.408	0.9
17_5	0.59386	0.91
34_6	0.71394	0.83
17_3, 34_4, 34_5	1	1

set3+3

Table 3.3: Linear correlation between I_2 (error in stay time) and I_4 (error in docking time)

planned as "critical". When the linear regression line has a coefficient zero, (instances 17_1, 17_2, 34_2, 34_3), there may be error in the stay time of some inbound trucks, but they do not cause any error in the docking time, *i. e.* those trucks are considered not critical. For the other instances for which the coefficient of the linear regression line is between 0 and 1, the situation is mixed: among the trucks staying longer than planned, some are critical and some are not.

From this observation, we draw the idea that finding ways to minimize the number of critical trucks would help improving the robustness of the IP model. We will come back to this idea in the next chapter.

3.4 PROPOSAL OF ROBUSTNESS METRICS

In section 3.3, we show that the curves linking the tolerance ε to the percentage off-limits are continuous and monotonic for two sources of variability: arrival time and unloading time. When the variability of truck arrival times increase, the tolerance to get a given percentage off-limits is also a strictly increasing curve when the percentage of trucks early or late increases. Based on these results, we propose a set of metrics to evaluate the robustness of a model subject to the three different sources of variability. Each robustness measure being a single numerical value, it is easier to exploit than a full set of data as represented for example in Figure 3.7. The robustness measures can be used to quantify how different IP models are able to absorb variations in transfer, unloading or truck arrival time.

ROBUSTNESS TO VARIABILITY IN TRANSFER TIME:

$$R^{\text{transfer}} = \text{tolerance } \varepsilon \text{ (in min) to get 10\% off-limits when } c_v^{\text{transfer}} = 0.1 \tag{3.1}$$

ROBUSTNESS TO VARIABILITY IN UNLOADING TIME:

$$R^{\text{unload}} = \text{tolerance } \varepsilon \text{ (in min) to get 10\% off-limits when } c_v^{\text{unload}} = 0.1 \tag{3.2}$$

ROBUSTNESS TO VARIABILITY IN TRUCK ARRIVAL TIME:

$$\begin{aligned} R^{\text{arrival}} &= \text{tolerance } \varepsilon \text{ (in min) to get 10\% off-limits} \\ &\text{when} \begin{cases} P(\sigma = +1) & = 20\% \\ P(\sigma = 0) & = 80\% \\ P(\sigma = -1) & = 0\% \end{cases} \quad \text{and } \delta = 10 \text{ min} \end{aligned} \tag{3.3}$$

Recall that tolerance ε corresponds to the value set for $\varepsilon_2 = \varepsilon_3 = \varepsilon_4 = \varepsilon_5$, as defined in section 3.2.2.3. The value of 10% off-limits is an arbitrary choice. When comparing two schedules, the idea of the indicator is to indicate whether the tolerance of one schedule is above or below the tolerance of the other. The indicator basically captures this information at a single state, which is 10% off-limits. The value does not seem unreasonable since 10% could be the maximum amount of trucks off-limit a platform can handle.

The values of c_v^{transfer} and c_v^{unload} are set to 0.1 because this value is the closest to industrial standard times, as shown already in section 3.2.2.1. A mean deviation of $\delta = 10$ minutes for late trucks seems a reasonable value, although we do not have industrial data to support this assumption.

Finally, the indicator on truck arrival times focuses on delayed trucks following the idea (mentioned in section 3.3.3) that delayed trucks have more impact on the schedule than early trucks.

If one wants to compare the robustness of two distinct scheduling models M1 and M2, subject to changes in truck arrival times, one shall use the simulation model to test both schedules and calculate the value of R^{arrival}. If $R^{\text{arrival}}(\text{M1}) < R^{\text{arrival}}(\text{M2})$, then M1 is more robust than M2 subject to changes in truck arrival times.

3.5 CONCLUSION

In order to know whether the models presented in chapter 2 are robust or not, this chapter proposes to use a discrete-event simulation model in order to submit the schedules to stochastic events. To answer the research question asked by Boysen and Fliedner [31], the acceptable level of uncertainty was shown to depend on the tolerance set by the operations manager. Based on a set of experiments, indicators are proposed to quantify the robustness of the models submitted to "reasonable" levels of uncertainty.

An extension of this work would be to test other sources of variations, *e.g.* uncertainties regarding the truck content. The possible correlations that can appear in practice between different sources of variations could also investigated.

The greedy algorithm used pallet routing in the simulation (algorithm 3.1) could also be used instead of the exact approaches (maximum flow) used in the tabu search (H3). Conversely, a maximum flow algorithm could be embedded in the simulation model (case (a) in Figure 3.1) in order to assess its performance when the pallets are transferred in an optimal way.

Different management policies for arriving trucks could be investigated – instead of being docked in a FIFO order, the trucks could be prioritized according to their punctuality, for example. The simulation model could then help comparing the different policies.

Another perspective is to use the simulation model and robustness indicators developed in order to design a simulation-optimization approach (case (c) in Figure 3.1 on page 70).

Is it possible to propose some models more robust than the models described in chapter 2 and evaluated in this chapter? To answer this question, the indicators developed will be used in chapter 4 to compare different robust versions of the original IP model.

L'incertitude est de tous les tourments
le plus difficile à supporter.

— Alfred de Musset

Chapter 4

ROBUST CROSSDOCK TRUCK SCHEDULING

This chapter proposes robust reformulations of the truck scheduling problem described in chapter 2. Reformulations are based on classical techniques in robust optimization (*minimax*, minimization of the expected regret) but also on techniques from robust project scheduling. Two different methods are used, resource redundancy and time redundancy. The robustness of the nine different models proposed is evaluated using the methodology and robustness indicators defined in chapter 3.

PLANIFICATION ROBUSTE DES CAMIONS

Dans ce chapitre, on cherche à proposer des reformulations plus robustes du modèle de planification de camions énoncé au chapitre 2. Les reformulations s'appuyent sur des techniques classiques d'optimisation robuste (*minimax* et minimisation du regret moyen), mais aussi sur des techniques issues d'un autre domaine, la planification de projets robustes. On distingue deux types de méthodes : celles qui assurent la robustesse par une redondance de ressources (peu pratiquée en gestion de projet puisque les ressources sont chères, mais adaptable à notre cas où les ressources sont les portes) et celles utilisant la redondance du temps, c'est-à-dire prévoyant des périodes-tampon pour compenser les aléas. Les modèles classiques d'optimisation robuste sont résolus en adaptant légèrement la recherche tabou du chapitre 2. Les trois modèles proposés qui utilisent la redondance de ressources sont résolus avec des versions adaptées d'IP* ou de H2, ainsi que deux modèles utilisant la redondance de temps. Les deux autres modèles ajoutant des périodes-tampons le font en post-traitement du résultat donné par IP* ou H2.

La robustesse de ces neuf modèles différents est évaluée à l'aide de la méthodologie et des trois indicateurs de robustesse définis au chapitre 3. Les résultats numériques permettent de montrer que la redondance de ressources, peu voire pas utilisée en gestion de projet à cause de son coût prohibitif, donne de très bons résultats une fois appliquée au cas du cross-docking.

4 ROBUST CROSSDOCK TRUCK SCHEDULING

Recall from chapter 3 the question asked by Boysen and Fliedner in their crossdock truck scheduling research agenda:

> "How to derive robust plans, *i. e.* plans which remain feasible in spite of (shorter) delays?"
>
> Boysen and Fliedner [31]

This chapter seeks to answer this question by proposing schedules that are robust to common levels of perturbations, *i. e.* that remain feasible (or can easily be fixed to become feasible again) when perturbations occur.

After a reminder of the problem's assumptions and a review of the literature regarding robustness in scheduling as well as robustness in cross-docking (section 4.1), we propose in section 4.2 different variations of the deterministic model described in chapter 2. They are compared and evaluated in section 4.3 using the indicators developed in chapter 3.

4.1 Robust truck scheduling with time windows: problem description

The model studied is similar to the one in chapter 2, but the realization of the schedule is now subject to uncertainties.

4.1.1 *Assumptions*

The platform considered is exactly the same as in chapter 2, thus all assumptions detailed in section 2.1.1 still hold.

The preferences of the transportation provider regarding the desired arrival and departure time for each truck are still expressed as time windows. The output of our model is then communicated back to the transportation provider that uses them as new references for the truck arrival and departure time. The difference with chapter 2 is that the schedule obtained is not necessarily executed exactly as planned. The trucks might actually arrive later than planned; the transfer or unloading processes might take longer, so that the assumption that product transfer is ensured in masked time does not hold any more.

When proposing robust schedules for the daily management of the platform, the perturbations considered should stay in a "reasonable" range, corresponding to discrepancies that can happen daily or weekly in the platform. Very big delays that occur in crisis situations, for instance a snowfall that paralyses all highway infrastructures in an entire region of France, are not taken into account in this study.

The three main sources of variability studied therefore take the following values, following the discussions in chapter 3:

- the *transfer time* follows a triangular distribution with $a = 2.67$, $b = 4.39$ and $m = 3.50$ minutes;
- the *unloading time* follows a triangular distribution with $a = 3.10$, $b = 5.10$ and $m = 4.10$ minutes;
- 20% of the trucks arrive late, and their delay follows an exponential distribution of parameter $\delta = 10$ minutes.

The objective of this chapter is to find robust solutions to our truck scheduling problem, *i. e.* solutions that are as close to optimal as possible for every possible situation that might occur. The robustness of each solution will be measured with the robustness indicators proposed in section 3.4.

The problem studied in this chapter therefore adds a robustness indicator to the performance measures used previously. The assumptions are summarized in Table 4.1.

	Strategical level				Tactical level				Operational level				Perf measures		
On which doors	Shape	Nb inbound doors	Nb outbound doors	Internal transport	Service mode	Pre-emption	Storage capacity	Resources capacity	Arrival time	Departure time	Truck filling	Interchangeability	Inventory level	Truck time deviation	Robustness
both	*	*	*	Manually	Exclusive	No	∞	lim.	/truck	Both	full	Dest.	✓	✓	✓

Table 4.1: Classification of the truck scheduling problem studied in chapter 4

4.1.2 *Robust scheduling: literature review*

In our context of mathematical programming for scheduling, adding robustness as a performance measure means changing the objective function in order to capture the robustness idea. It can be done in many different ways, reviewed by Sabuncuoglu and Goren [170] in their review focusing on robustness and stability in a manufacturing environment. They propose an organized list of different objective functions used to ensure stability and robustness. Based on their work and after adding other measures proposed in more recent papers, we can list (not exhaustively) the main possible objective functions for robust scheduling.

OBJECTIVE FUNCTIONS BASED ON REALIZED PERFORMANCE. The idea is to ensure that the performance level achieved by the schedule stays high when facing a disruption. For a minimization problem, this can be done by minimizing the expected realized performance, minimizing the worst-case performance (*minimax method*: the worst-case performance is the max of the performances obtained for all the scenarios considered; this criteria is called *absolute robustness* by Kouvelis and Yu [114]), minimizing the performance of the schedule in the most probable scenario, minimizing the expected deviation of the realized schedule's performance from the initial deterministic performance, minimizing the variance of realized performance measure...

objective functions based on regret. We call regret the difference between the realized and the optimal performance, *i. e.* the performance that would have been realized if the disruptions were known in advance and used as data. The idea is to ensure that the performance level achieved is close to what it would have been with a full information. It is usually done by minimizing the expected regret, or minimizing the regret in the worst case (*minimax regret method*; this criteria is called *absolute deviation* or *relative deviation* by Kouvelis and Yu [114]).

objective functions based on slacks. These measures are proposed by Hazır *et al.* [101] in the context of robust project scheduling. They are based on the slack of some project tasks, *i. e.* the amount of delay that a task can take without delaying the completion time of the total project. A slack is therefore a buffer time that can protect a specific task against delay or disruptions, when placed right after the task in a Gantt chart. Using a simulation experiment, Hazır *et al.* show that two performance measures have a high correlation with indicators on the project punctuality: the maximum weighted slack (where the weight of a slack is the number of immediate successors, in the Gantt chart, of the task protected by the slack, or its total number of successors), and the maximum ratio between the total project buffer size and the total project completion time.

Objective functions based on realized performance and based on regret are not specific to robust scheduling, and are largely used in robust optimization in general. The interested reader can refer, for example, to Nikulin [156] for an extended annotated bibliography of robustness in combinatorial optimization and scheduling theory, or to Gabrel *et al.* [74] for a more recent review of the literature regarding robust optimization.

Slack-based measures, on the contrary, are very specific to project scheduling. They follow the idea emphasized by Davenport and Beck [55], who show that redundancy-based techniques are a way to proactively ensure the robustness of a schedule. For slack-based indicators, the redundancy is applied on *time*, since the idea is to keep reserve time or buffer time periods. Davenport and Beck [55] note that *resource* redundancy (keeping some resources in standby) is another way to ensure robustness in scheduling. However, resource redundancy is not usually used in project management, since keeping idle resources would be unreasonably expensive.

Time redundancy is much more frequently used in project scheduling. It has been originally proposed in 1990 by Chiang and Fox [45] (and later Gao [76]) who developed the concept of temporal protection. The "protected" duration of each activity equals its original duration augmented with the duration of breakdowns that are expected to occur during the activity execution, based on breakdown statistics from the resources. The schedule is obtained by solving the scheduling problem in which the task durations are the protected ones. Similarly, Mehta and Uzsoy [147] insert additional idle time into the predictive schedule to absorb the impact of machine breakdowns. The insertion is done as a post-treatment of a sequence obtained by a heuristic. Davenport *et al.* [56] propose improvements of this temporal protection technique with their *time window slack* and *focused time window slack* approaches. Slacks are not included into the activity duration, but explicitly computed per activity in solution schedules. In this way, the same slack time can protect more than one activity, and slacks can be concentrated in the areas of the schedule that are the most crucial. In [201], van de Vonder *et al.* propose different algorithms to include time buffers in a project schedule: the *virtual activity duration extension* in which the time buffer depends on the variability in the activity durations of the predecessors, and the *starting time criticality* in which the time buffers depend on both the weights of the activities and the variance of the activities duration. The heuristic adds time buffers in front of the most critical activities until adding more safety would no longer improve stability. They propose local search improvements, with a specific algorithm combining steepest and fastest descent, and a tabu search. They experimentally show that the *starting time criticality* heuristic performs best.

Fuzzy set theory can also be used to determine the size of the buffer; see *e.g.* Li and Chen [121].

The different techniques detailed here are either generic robust optimization techniques, or techniques that are specific to scheduling or project scheduling. In the next section, the cross-docking literature is reviewed to see which papers deal with robustness, and which of the techniques presented above are actually used in the cross-docking context.

4.1.3 *Robustness in the cross-docking literature*

Sensitivity analysis checks the effect of parameter changes.

In their review of the scheduling and project scheduling literature, Herroelen and Leus [103] identify different strategies used to cope with uncertainty: reactive scheduling, stochastic scheduling, fuzzy scheduling, proactive robust scheduling and sensitivity analysis. Keeping the sensitivity analysis aside, we use the same classification to order the articles mentioned in this section. They are a subset of the articles mentioned in chapter 1 that address robustness, or more broadly speaking any sort of uncertainty on the input data.

4.1.3.1 *Reactive scheduling*

Reactive scheduling aims at revising or re-optimizing the schedule when an unexpected event occurs. It is often called "on-line scheduling" in the cross-docking literature.

Wang and Regan [212] solve a truck-to-door assignement problem in which the arrivals of inbound trucks follow an exponential distribution, and all other process times are deterministic. Their algorithm, using real-time information about freight transferring within the crossdock, chooses on-line the best truck to be placed at each door that becomes available. The robust truck-to-door assignment problem is also dealt with by Yu *et al.* [218], who propose an online policy when most of the data regarding the inbound trucks (number, arrival time, contents, unloading time) is uncertain. Acar *et al.* [2] also propose a dynamic heuristic to assign the trucks to the docks in real time. Larbi *et al.* [115] schedule the transshipment of pallets in a single receiving door and a single shipping door crossdock where preemption is allowed, with partial and no information on the sequence of upcoming trucks. In the case of no information, only the daily quantities for each destination are supposed to be known. The problem is solved with a heuristic based on a probabilistic decision rule: after an inbound truck has been unloaded, the outbound truck which has the highest probability to be fully loaded with the minimum expected cost is placed at the outbound door. In the case of partial information, only the sequence of the next Z inbound trucks and their contents are known. Two different approaches are presented: first, applying on a rolling horizon the optimal algorithm developed for the full-information case; second, a heuristic that hybrids the full-information and the no-information methods.

Reactive scheduling can be seen as a way to fix a schedule when unexpected events occur, it is an *a posteriori* approach. It is, for example, the role of the greedy algorithm which is in charge of the pallet transfer in the simulation (see algorithm 3.1 in section 3.1.2). The goal in this chapter is rather to make the schedule robust *a priori*, or in a "proactive" way as named by Herroelen and Leus [103].

4.1.3.2 *Stochastic scheduling*

Stochastic scheduling aims at minimizing the expected objective value, which implies that the probability distributions of the uncertain data are known.

Beside their on-line policy for the truck-to-doors assignment problem, Yu *et al.* [218] propose a scenario-based stochastic model in order to assign fixed destinations to the outbound doors, given the on-line policy. The objective is to minimize the expected workload (total distance walked by the worker), taking the random variation of freight volumes into account. The problem is solved by local search and genetic algorithm.

Yu *et al.* are the only authors who use stochastic programming for crossdock scheduling, more precisely to allocate destinations to doors. A possible explanation for the scarcity of stochastic optimization in cross-docking resides in the fact that the probability distributions of uncertain data can be hard to obtain in the industrial context. In our case, we assume some probability distributions in order to test the model; but using them as input data in our optimization might make the problem too hard to be solved.

4.1.3.3 *Fuzzy scheduling*

Fuzzy scheduling involves imprecision rather than uncertainty; instead of probability distributions that are not always easy to obtain, the uncertain data are modeled with fuzzy numbers. For more information on fuzzy scheduling, the interested reader can refer to Stefanini *et al.* [188].

However, the precise form of a fuzzy number is difficult to describe by an expert. It might be why no articles dealing with cross-docking operations make use of this method.

4.1.4 *Proactive robust scheduling*

Within the proactive robust scheduling method, we can distinguish, as done already in section 4.1.2, between generic robust optimization techniques and redundancy-based techniques.

Four articles among the ones listed in chapter 1 use robust optimization techniques. Bozer and Carlo [34] propose a model for inbound and outbound door assignments in crossdocks, but notice that it can create large variations in the workload from one night to the next. In order to reduce those variations, they solve two different problems: minimizing the total workload (*minisum*) and minimizing the maximum workload (*minimax*). A solution is called robust if the workload of the worst night in the *minisum* solution is close to the workload in the *minimax* solution. Werners and Wülfing [214] propose a model to schedule the outbound trucks in a parcel sorting center, minimizing the total transportation effort. In order to achieve robustness, they minimize the maximal regret of four different scenarios corresponding to different activity levels in the facility. They also ensure that the transportation effort in the robust solution is close to every scenario-optimal objective value. These two articles focus on the activity level within the platform. To find articles closer to the work presented in this chapter, one has to refer to Konur and Golias [112, 113] who deal with the inbound truck-to-door scheduling in a crossdock, where the truck arrival times are uncertain (modeled by a triangular distribution). They solve the problem for deterministic, optimistic (the total waiting times is expected to be minimum), pessimistic (the total waiting times is expected to be maximum) and hybrid cases. The hybrid approach is shown to outperform the others in certain cases. The work by Konur and Golias focuses on the inbound trucks only and aims at minimizing the total waiting time of the trucks. In this chapter, we also use robust optimization techniques but for inbound and outbound sides, with different performance measures.

Hybrid case: the expected truck arrival times are the arithmetic average of the expected truck arrival times given by the pessimistic and optimistic approaches.

Time redundancy techniques are only used in the crossdock operations context by Acar *et al.* [2]. They aim at minimizing the variance of the doors' idle times, in order to spread the inbound trucks on a given dock as evenly as possible and thus create buffer times between trucks. Their technique will be adapted to our case and compared to other robust versions of the model.

4.2 ROBUST VERSIONS OF THE INITIAL PROBLEM

Among the various methods detailed in the literature review, four are used to propose robust versions of our problem.

Two are based on widely used robust optimization techniques and prefixed with the letter R: the *minimax* method (R1) and the minimization of expected regret (R2).

The other two methods are derived from project scheduling and adapted to the cross-docking context: resource redundancy (too expensive and therefore barely used for project scheduling, but easy to adapt to our case) and time redundancy. The resources that can be made redundant are the doors in our case. Therefore the models implementing this method are prefixed with the letter D, while the models implementing time redundancy are prefixed with T.

Figure 4.1 on the following page shows with a small case (3 inbound and 3 outbound doors) how an original schedule (a) can be modified to ensure resource redundancy (b) or time redundancy (c). In case (b), the trucks are all grouped on two doors so that the third one stays constantly available as a buffer door; it can process any arriving truck if the operations get delayed. In case (c), time buffers are inserted in between the different trucks to avoid delays propagation. In both cases, the truck arrival and departure times are not modified too much compared to the initial schedule (a).

Depending on the nature of the problem, different solution strategies are used. The different robust versions and their resolution methods are summarized in Figure 4.2 on the next page, and detailed in the following sections.

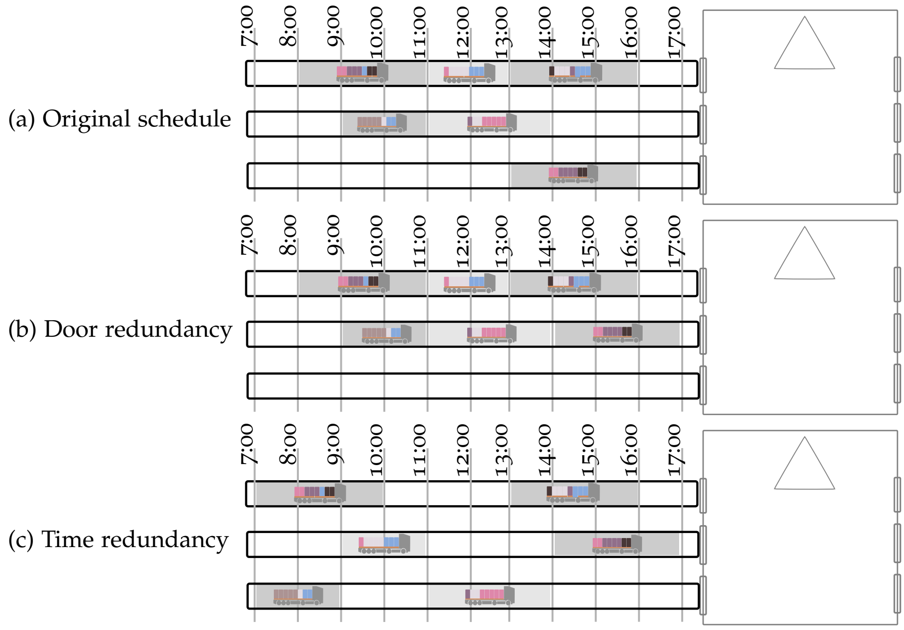


Figure 4.1: Two ways of adding robustness

Robust optimization		Project scheduling	
Min objective in the worst case	Min expected regret	Resource redundancy	Time redundancy
R1: minimax	R2: min $\Pi_0(S) - \Pi_0(S_0)$	D1: min the door occupation ratio	T1: insert buffers of equal length
		D2: min nb of doors used every hour	T2: insert buffers of length prop. to nb successors
		D3: min nb of critical trucks	T3: min buffer lengths standard deviations
			T4: max buffer lengths weighted sum

Solution methods
- IP* or H1/H2
- Tabu search H3
- IP* or H1/H2 + post-treatment

Figure 4.2: Summary of the robust versions

4.2.1 *Robust optimization techniques*

In most papers using robust optimization such as *minimax* or the minimization of expected regret, local searches (generally tabu searches or genetic algorithms) are used to solve the robust counterparts of the models. In our case, the tabu search described in section 2.3.3 (H3) can easily be adapted by changing the evaluation of the objective function to match the cases displayed below.

4.2.1.1 R1*: minimax method*

The minimax method consists in minimizing worst case performance. Applied to our problem, the minimax model is formulated as shown below.

min	Ω	
s.t.	$\Omega \geq \Pi_0(s)$	$\forall$ scenario s in the set of scenarios considered
	other constraints of IP*	

R1

The set of scenarios considered represents different cases where the trucks are delayed. Similar to chapter 3, the scenarios chosen have 20% of trucks late, with delays following an exponential distribution of parameter $\delta = 1$ min, 5 min, 10 min, 15 min and 30 min.

4.2.1.2 R2*: minimizing the expected regret*

The objective in this case is to minimize the expected deviation between the realized schedule's performance (here when 20% of the trucks are delayed) and the performance of the deterministic scenario (noted S_0). The problem is formulated as shown below.

min	$\Pi_0(S_{20\%\ \text{delayed}}) - \Pi_0(S_0)$
s.t.	constraints of IP*

R2

4.2.2 *Resource redundancy*

The objective of resource redundancy is to ensure that another resource will be available to execute a job when disruptions occur. This solution is not often used in project scheduling, because resources are expensive: it is financially more interesting to plan a longer project than to pay people to stay idle. In cross-docking however, the resources (doors) are not necessarily very expensive, especially in big platforms that do not always use all their doors.

Various strategies are therefore developed in this section, aiming at using less than the total number of doors available.

4.2.2.1 *Model* D1

The goal of this model is to minimize the door occupation ratio. Recall from Equation 2.1 section 2.2.4 that ratio R is the average number of trucks present per door and per hour. This definition is adapted here in order to define an inbound and an outbound ratio. Denoting by n_h^I (resp. n_h^O) the number of inbound (resp. outbound) trucks docked at time $h \in \mathcal{H}$, the inbound ratio R^I and outbound ratio R^O are defined as follows:

$$R^I = \frac{\sum_{h\in\mathcal{H}} n_h^I}{N^I|\mathcal{H}|} \qquad R^O = \frac{\sum_{h\in\mathcal{H}} n_h^O}{N^O|\mathcal{H}|} \tag{4.1}$$

The number of inbound and outbound trucks docked at time h can be expressed for all $h \in \mathcal{H}$ using their presence time windows:

$$n_h^I = \sum_{i\in I}\sum_{k\in K_i} W_{ikh}^I w_{ik}^I \qquad n_h^O = \sum_{o\in O}\sum_{k\in K_o} W_{okh}^O w_{ok}^O \tag{4.2}$$

Reducing the door occupation ratio is likely to leave more free doors, which ensures resource redundancy. The objective of model D1 is therefore to minimize ratios R^I and R^O.

Adding another weighted penalty to the objective function would make the weight setting difficult – how can the importance of the original objectives be compared relatively to the new objective of minimizing ratio R? To circumvent this issue, the different objectives are solved in lexicographic order. IP* is first solved to find the optimal value of the different parts of the objective function (Π_0^α, Π_0^β, Π_0^γ). Then the model noted $(\text{IP*})^{\text{D1}}$ finds among the optimal solutions, the solution whose ratio $R^I + R^O$ is the smallest. $(\text{IP*})^{\text{D1}}$ is formulated using two new decision variables, r^I and r^O, representing the ratios defined in Equation 4.1.

$(\text{IP*})^{\text{D1}}$ has a few differences with IP*: the objective function aims at minimizing the ratio of trucks present at the door. Constraints $(1)^{\text{D1}}$ to $(3)^{\text{D1}}$ ensure that the different elements of IP* objective function stay equal to their optimal value calculated before. Constraints $(15)^{\text{D1}}$ and $(16)^{\text{D1}}$ define r^I and r^O using a combination of Equation 4.1 and Equation 4.2.

In the case of larger instances, we need to use the heuristics. To adapt the above method to heuristics H1 and H2, we use this lexicographic objective for both steps of the heuristic. Similar modifications are brought to the IP models composing the heuristics. For example, H1 is solved as follows:

- Solve IP1 as before (model on page 55).
- Solve $(\text{IP1})^{\text{D1}}$ (see Appendix C).
- Run IP^{*1} as before (see Appendix C).
- Run $(\text{IP}^{*1})^{\text{D1}}$ (see Appendix C).

$$
\begin{array}{lll}
\min & r^I + r^O & \\
& & \\
\text{s.t.} & \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} p^I_{ik} w^I_{ik} \leq \Pi^\alpha_0 & (1)^{\text{D1}} \\
& \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} p^O_{ok} w^O_{ok} \leq \Pi^\beta_0 & (2)^{\text{D1}} \\
& \sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic} \leq \Pi^\gamma_0 & (3)^{\text{D1}} \\
& \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} W^I_{ikh} w^I_{ik} \leq N^I \quad \forall h \in \mathcal{H} & (4) \\
& \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} W^O_{okh} w^O_{ok} \leq N^O \quad \forall h \in \mathcal{H} & (5) \\
& x_{hio} + \sum_{c\in\mathcal{C}} s^I_{hic} \leq F \sum_{k\in\mathcal{K}_i} W^I_{ikh} w^I_{ik} \quad \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O} & (6) \\
& x_{hio} + s^O_{ho} \leq F \sum_{k\in\mathcal{K}_o} W^O_{okh} w^O_{ok} \quad \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O} & (7) \\
& \sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co} x_{hio} + \sum_{h\in\mathcal{H}} s^I_{hic} = Q_{ic} \quad \forall i \in \mathcal{I}, c \in \mathcal{C} & (8) \\
& \sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s^O_{ho} = F \quad \forall o \in \mathcal{O} & (9) \\
& \sum_{o\in O} x_{hio} + \sum_{c\in\mathcal{C}} s^I_{hic} \leq M \quad \forall i \in \mathcal{I}, h \in \mathcal{H} & (10) \\
& \sum_{k\in\mathcal{K}_i} w^I_{ik} = 1 \quad \forall i \in \mathcal{I} & (11) \\
& \sum_{k\in\mathcal{K}_o} w^O_{ok} = 1 \quad \forall o \in \mathcal{O} & (12) \\
& s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s^I_{hic} - \sum_{o\in\mathcal{O}} Z_{co} s^O_{ho} \quad \forall c \in \mathcal{C}, h \in \mathcal{H} \setminus \{0\} & (13) \\
& s_{0c} = \sum_{i\in\mathcal{I}} s^I_{0ic} - \sum_{o\in\mathcal{O}} Z_{co} s^O_{0o} \quad \forall c \in \mathcal{C} & (14) \\
& \sum_{h\in\mathcal{H},i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} W^I_{ikh} w^I_{ik} \leq r^I N^I |\mathcal{H}| & (15)^{\text{D1}} \\
& \sum_{h\in\mathcal{H},o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} W^O_{okh} w^O_{ok} \leq r^O N^O |\mathcal{H}| & (16)^{\text{D1}} \\
& & \\
& x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc}, r^I, r^O \in \mathbb{N}^+ \quad \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O}, c \in \mathcal{C} & \\
& w^I_{ik}, w^O_{ok} \in \{0,1\} \quad \forall i \in \mathcal{I}, o \in \mathcal{O}, k \in K &
\end{array}
$$

(IP*)$^{\text{D1}}$

4.2.2.2 *Model* D2

Model D2 aims at minimizing the number of doors used every hour. D2 uses an idea similar to D1, but instead of minimizing the ratio in general, the model minimizes the number n_h^I (resp. n_h^O) of inbound (resp. outbound) doors used at every hour $h \in H$: the inequalities $(15)^{D1}$ and $(16)^{D1}$ are thus defined for every hour instead of a sum over all hours of the horizon. Similarly to D1, the decision variables noted n_h^I and n_h^O are introduced in an optimization model $(\text{IP}^*)^{D2}$, called in lexicographic order after IP*, which determines the values of Π_0^α, Π_0^β and Π_0^γ.

When heuristics are to be used for larger instances, the solution strategy is also similar to the one described for D1.

$$
\begin{array}{lllr}
\min & \sum_{h\in\mathcal{H}} n_h^I + n_h^O & & \\
 & & & \\
\text{s.t.} & \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} p_{ik}^I w_{ik}^I \leq \Pi_0^\alpha & & (1)^{D2} \\
 & \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} p_{ok}^O w_{ok}^O \leq \Pi_0^\beta & & (2)^{D2} \\
 & \sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s_{hic}^I \leq \Pi_0^\gamma & & (3)^{D2} \\
 & \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} W_{ikh}^I w_{ik}^I \leq N^I & \forall h \in \mathcal{H} & (4) \\
 & \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} W_{okh}^O w_{ok}^O \leq N^O & \forall h \in \mathcal{H} & (5) \\
 & x_{hio} + \sum_{c\in\mathcal{C}} s_{hic}^I \leq F \sum_{k\in\mathcal{K}_i} W_{ikh}^I w_{ik}^I & \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O} & (6) \\
 & x_{hio} + s_{ho}^O \leq F \sum_{k\in\mathcal{K}_o} W_{okh}^O w_{ok}^O & \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O} & (7) \\
 & \sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co} x_{hio} + \sum_{h\in\mathcal{H}} s_{hic}^I = Q_{ic} & \forall i \in \mathcal{I}, c \in \mathcal{C} & (8) \\
 & \sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s_{ho}^O = F & \forall o \in \mathcal{O} & (9) \\
 & \sum_{o\in O} x_{hio} + \sum_{c\in\mathcal{C}} s_{hic}^I \leq M & \forall i \in \mathcal{I}, h \in \mathcal{H} & (10) \\
 & \sum_{k\in\mathcal{K}_i} w_{ik}^I = 1 & \forall i \in \mathcal{I} & (11) \\
 & \sum_{k\in\mathcal{K}_o} w_{ok}^O = 1 & \forall o \in \mathcal{O} & (12) \\
 & s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s_{hic}^I - \sum_{o\in\mathcal{O}} Z_{co} s_{ho}^O & \forall c \in \mathcal{C}, h \in \mathcal{H} \setminus \{0\} & (13) \\
 & s_{0c} = \sum_{i\in\mathcal{I}} s_{0ic}^I - \sum_{o\in\mathcal{O}} Z_{co} s_{0o}^O & \forall c \in \mathcal{C} & (14) \\
 & \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} W_{ikh}^I w_{ik}^I \leq n_h^I & \forall h \in H & (15)^{D2} \\
 & \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} W_{okh}^O w_{ok}^O \leq n_h^O & \forall h \in H & (16)^{D2} \\
 & & & \\
 & x_{hio}, s_{hic}^I, s_{ho}^O, s_{hc}, n_h^I, n_h^O \in \mathbb{N}^+ & \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O}, c \in \mathcal{C} & \\
 & w_{ik}^I, w_{ok}^O \in \{0,1\} & \forall i \in \mathcal{I}, o \in \mathcal{O}, k \in K &
\end{array}
$$

$(\text{IP}^*)^{D2}$

4.2.2.3 *Model* D3

Model D3 aims at minimizing the number of critical trucks. As observed experimentally in section 3.3.4, a key role is played in the operations' punctuality by some trucks we call *critical trucks*, after the project management term of "critical tasks". A critical task in project management is a task that does not have flexibility, *e.g.* that delays the entire project if the task is delayed (see *e.g.* Project Management Institute [163]). Similarly, we define a critical truck as one that, when late, propagates its delay to the next arriving trucks. When a truck i_1 arrives at the platform, it can be docked at one of the doors that are available at that time. If it has no choice but to dock at a door that was just freed by a truck i_0, we call i_0 critical. Indeed, if i_0 is late, i_1 will have to wait before docking.

The number of critical trucks is formally defined in Definition 1.

Definition 1. Critical trucks. We denote by n_h^I (resp. n_h^O) the number of inbound (resp. outbound) trucks docked at time $h \in \mathcal{H}$, and by $m_h^I \leq N^I$ (resp. $m_h^O \leq N^O$) the number of inbound (resp. outbound) trucks coming in at time $h > 0$. The number of critical inbound (resp. outbound) trucks leaving at time $h \geq 0$ is:

$$c_h^I = \max(0, m_h^I + n_{h-1}^I - N^I)$$

$$c_h^O = \max(0, m_h^O + n_{h-1}^O - N^O)$$

No truck leaves at time $h = 0$, so $c_0^I = c_0^O = 0$.

New decision variables c_h^I (resp. c_h^O) are added to represent the number of critical inbound (resp. outbound) trucks leaving at time $h \in \mathcal{H}$, that should be minimized in D3.

In order to express the number of critical trucks as defined in Definition 1, a new data element is needed:

A_{kh}^I = 1 if hour $h \in \mathcal{H}$ is the starting time of slot $k \in \mathcal{K}_i$ ($i \in \mathcal{I}$);
A_{kh}^O = 1 if hour $h \in \mathcal{H}$ is the starting time of slot $k \in \mathcal{K}_o$ ($o \in \mathcal{O}$).

The value of matrices A^I and A^O can easily be derived from the values of W^I and W^O. Using this new data, the number of inbound (resp. outbound) trucks arriving to dock at time $h \in \mathcal{H}$ can be written as:

$$m_h^I = \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} A_{kh}^I w_{ik}^I \qquad m_h^O = \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}_o} A_{kh}^O w_{ok}^O \tag{4.3}$$

Model $(\text{IP}^*)^{\text{D3}}$ can thus be formulated as shown on the next page. Constraints $(1)^{\text{D3}}$ to $(3)^{\text{D3}}$ ensure that the values of Π_0^α, Π_0^β and Π_0^γ stay consistent with the first step of the optimization. Constraints $(17)^{\text{D3}}$ and $(18)^{\text{D3}}$ define the number of critical trucks as defined in Definition 1. Constraints $(15)^{\text{D3}}$ and $(16)^{\text{D3}}$ define the number of

$$
\begin{array}{lllr}
\min & \sum_{h\in H} c_h^I + c_h^O & & \\
\text{s.t.} & \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} p_{ik}^I w_{ik}^I \le \Pi_0^\alpha & & (1)^{\text{D3}} \\
& \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} p_{ok}^O w_{ok}^O \le \Pi_0^\beta & & (2)^{\text{D3}} \\
& \sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s_{hic}^I \le \Pi_0^\gamma & & (3)^{\text{D3}} \\
& \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} W_{ikh}^I w_{ik}^I \le N^I & \forall h\in\mathcal{H} & (4) \\
& \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} W_{okh}^O w_{ok}^O \le N^O & \forall h\in\mathcal{H} & (5) \\
& x_{hio} + \sum_{c\in\mathcal{C}} s_{hic}^I \le F\sum_{k\in\mathcal{K}_i} W_{ikh}^I w_{ik}^I & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (6) \\
& x_{hio} + s_{ho}^O \le F\sum_{k\in\mathcal{K}_o} W_{okh}^O w_{ok}^O & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (7) \\
& \sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co} x_{hio} + \sum_{h\in\mathcal{H}} s_{hic}^I = Q_{ic} & \forall i\in\mathcal{I}, c\in\mathcal{C} & (8) \\
& \sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s_{ho}^O = F & \forall o\in\mathcal{O} & (9) \\
& \sum_{o\in O} x_{hio} + \sum_{d\in D} s_{hid}^I \le M & \forall i\in\mathcal{I}, h\in\mathcal{H} & (10) \\
& \sum_{k\in\mathcal{K}_i} w_{ik}^I = 1 & \forall i\in\mathcal{I} & (11) \\
& \sum_{k\in\mathcal{K}_o} w_{ok}^O = 1 & \forall o\in\mathcal{O} & (12) \\
& s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s_{hic}^I - \sum_{o\in\mathcal{O}} Z_{co} s_{ho}^O & \forall c\in\mathcal{C}, h\in\mathcal{H}\setminus\{0\} & (13) \\
& s_{0c} = \sum_{i\in\mathcal{I}} s_{0ic}^I - \sum_{o\in\mathcal{O}} Z_{co} s_{0o}^O & \forall c\in\mathcal{C} & (14) \\
& \sum_{i\in I}\sum_{k\in\mathcal{K}_i} W_{ikh}^I w_{ik}^I \le n_h^I & \forall h\in\mathcal{H} & (15)^{\text{D3}} \\
& \sum_{o\in O}\sum_{k\in\mathcal{K}_o} W_{okh}^O w_{ok}^O \le n_h^O & \forall h\in\mathcal{H} & (16)^{\text{D3}} \\
& c_h^I \ge \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} A_{kh} w_{ik}^I + n_{h-1}^I - N^I & \forall h\in\mathcal{H}\setminus\{0\} & (17)^{\text{D3}} \\
& c_h^O \ge \sum_{o\in O}\sum_{k\in\mathcal{K}_o} A_{kh} w_{ok}^O + n_{h-1}^O - N^O & \forall h\in\mathcal{H}\setminus\{0\} & (18)^{\text{D3}} \\
& & & \\
& x_{hio}, s_{hic}^I, s_{ho}^O, s_{hc}, n_h^I, n_h^O, c_h^I, c_h^O \in \mathbb{N}^+ & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}, c\in\mathcal{C} & \\
& w_{ik}^I, w_{ok}^O \in \{0,1\} & \forall i\in\mathcal{I}, o\in\mathcal{O}, k\in K &
\end{array}
$$

$(\text{IP}^*)^{\text{D3}}$

trucks docked at time $h \in \mathcal{H}$ as defined in Equation 4.2. Other constraints are the same as those in IP*.

The solution strategy for small instances and larger ones is again similar to the one described for D1: the models are solved in lexicographic order, calculating first that the values of Π_0^α, Π_0^β and Π_0^γ, then minimizing the number of critical trucks using $(\text{IP*})^{\text{D3}}$.

4.2.3 *Time redundancy*

Time redundancy methods aim at adding buffer time periods (or *slacks*) in the schedule in order to ensure that no truck is critical. Since time redundancy techniques are broadly used in the project management literature, the different solutions presented here are derived from ideas already mentioned in the literature review (section 4.1.2).

4.2.3.1 *Post-treatment* T1

An idea very simple to understand for the manager and easy to implement is to insert buffers of equal lengths between the presence slots of the trucks at the doors. It can be done by adapting the project scheduling techniques for inserting buffer developed by Mehta and Uzsoy [147].

Buffers are inserted by a post-treatment of the schedule (calculated by IP* or H1, H2 or H3 as detailed in chapter 2). The planning horizon should not be extended, therefore the goal is not to add extra time but to redistribute the free time available in the planning.

T1 *inserts buffers of equal length.*

The total buffer available on the planning horizon is divided equally among all trucks as show in algorithm 4.1 on the following page. Since T1 is a greedy post-treatment heuristic, it cannot ensure the coordination between inbound and outbound trucks. Because the truck presence time windows are only shifted by a small amount of time compared to the solution calculated with the IP model, the solution is likely to stay feasible. However, this cannot be guaranteed.

4.2.3.2 *Post-treatment* T2

T2 consists in inserting buffers, similar to T1; but in this version the length of the buffer inserted after a specific truck is proportional to the number of successors, as suggested by Hazır *et al.* [101]. The number of successors of a truck $i \in \mathcal{I}$ (resp. $o \in \mathcal{O}$) is the number of trucks that come at the same door after truck i (resp. o). When its number of successors is bigger, a given truck is more likely to be critical and to propagate a delay. A bigger buffer is thus allocated to the trucks with the bigger number of successors, in order to avoid propagating the delays to its successors.

T2 *inserts buffers of length proportional to its number of successors.*

The insertion is made as a post-treatment, similar to T1. The insertion heuristic is detailed in algorithm 4.2 on the next page.

1. Run IP* for small instances, or H2 for larger instances.
2. Following a FIFO policy, match each inbound truck to an inbound door and each outbound truck to an outbound door.
3. Calculate the total inbound (resp. outbound) buffer size β_d^I (resp. β_d^O) at each inbound door d^I (resp. outbound door d^O).
4. Insert a buffer of length $\frac{\beta^I}{|\mathcal{I}|}$ after each truck $i \in \mathcal{I}$, *i. e.* move the next truck arriving at the same door so that the time between them is exactly $\frac{\beta^{\mathcal{I}}}{|\mathcal{I}|}$. Similarly, insert a buffer of length $\frac{\beta^O}{|\mathcal{O}|}$ after each outbound truck $o \in \mathcal{O}$.

Algorithm 4.1: Inserting buffers of equal lengths between successive trucks

1. Run IP* for small instances, or H2 for larger instances.
2. Following a FIFO policy, match each inbound truck to an inbound door and each outbound truck to an outbound door.
3. Calculate the total inbound (resp. outbound) buffer size β_d^I (resp. β_d^O) at each inbound door d^I (resp. d^O).
4. Calculate the number of successors σ_i^I of each inbound truck $i \in \mathcal{I}$, and the number of successors σ_o^O of each outbound truck $o \in \mathcal{O}$.
5. Insert a buffer of length $\frac{\beta^I \times \sigma_i^I}{\sum_{i \in I} \sigma_i^I}$ after each truck $i \in \mathcal{I}$, *i. e.* move the next truck arriving at the same door so that the time between them is exactly the calculated length. Similarly, insert a buffer of length $\frac{\beta^O \times \sigma_o^O}{\sum_{o \in \mathcal{O}} \sigma_o^O}$ after each truck $o \in \mathcal{O}$.

Algorithm 4.2: Inserting buffers of lengths proportional to the number of successors

4.2.3.3 *Model* T3

Model T3 implements the idea proposed by Acar *et al.* [2]. Their objective is to minimize the standard deviation of the buffer lengths, *i. e.* to ensure that the buffer lengths tend to be similar.

New data and new decision variables must be added to IP* in order to model the buffer lengths explicitly. The set of possible buffers, including the buffer of length 0, is denoted by $\mathcal{L}$.

B_{lh} $= 1$ if buffer $l \in \mathcal{L}$ includes hour $h \in \mathcal{H}$, 0 otherwise.
D_l duration of buffer $l \in \mathcal{L}$ (in hours).
b^I_{il} $= 1$ if buffer $l \in \mathcal{L}$ is chosen to protect inbound truck $i \in \mathcal{I}$, 0 otherwise.
b^O_{ol} $= 1$ if buffer $l \in \mathcal{L}$ is chosen to protect outbound truck $o \in \mathcal{O}$, 0 otherwise.

Because a buffer protects a truck by being placed directly after the truck departure time, buffers are closely related to the presence time windows selected for the trucks. We therefore denote by $\mathcal{L}_k$ the subset of $\mathcal{L}$ that only includes buffers starting right after the ending time of slot k.

T3 minimizes the deviation between the buffer length and their average length.

The standard deviation is not a linear function, therefore the objective function must be adapted to be solved with an IP model. The idea of Acar *et al.* [2] is to unify the buffer lengths as much as possible. To reach a comparable aim, the choice is made to minimize the difference between the length of each selected buffer and the average length of all buffers, denoted avg.

Model (IP*)^{T3} is written as shown on the following page. Absolute values appear in the objective function, to define the gap between the length of a particular buffer and the average buffer length. Note that a function of the form $z = |x - y|$ can be linearized as follows if z appears in the minimization objective:

$$z = |x - y| \Leftrightarrow \begin{cases} z \geq x - y \\ z \geq y - x \end{cases} \tag{4.4}$$

The average buffer length used in the objective function is defined by constraint (17)^{T3}. Constraints (4)^{T3} and (5)^{T3} replace constraints (4) and (5) of IP*, adding buffers in between the truck presence time windows. Constraints (15)^{T3} and (16)^{T3} make sure that each truck is protected by exactly one buffer.

Similar to the other IP models presented in the previous sections, model (IP*)^{T3} is used in lexicographic order after running IP*. Constraints (1)^{T3} to (3)^{T3} ensure that the different elements of the objective function stay within the limits defined in the first step of the optimization.

$$
\begin{array}{llll}
\min & \sum_{i\in\mathcal{I}} \left|avg - \sum_{l\in\mathcal{L}} D_l b^I_{il}\right| + \sum_{o\in\mathcal{O}} \left|avg - \sum_{l\in\mathcal{L}} D_l b^I_{il}\right| & & \\
\text{s.t.} & \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} p^I_{ik} w^I_{ik} \leq \Pi^\alpha_0 & & (1)^{\text{T3}} \\
& \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} p^O_{ok} w^O_{ok} \leq \Pi^\beta_0 & & (2)^{\text{T3}} \\
& \sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic} \leq \Pi^\gamma_0 & & (3)^{\text{T3}} \\
& \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} (W^I_{ikh} w^I_{ik} + \sum_{l\in\mathcal{L}_k} B_{lh} b^I_{il}) = N^I & \forall h \in \mathcal{H} & (4)^{\text{T3}} \\
& \sum_{o\in\mathcal{O}} \sum_{k\in\mathcal{K}_o} (W^O_{okh} w^O_{ok} + \sum_{l\in\mathcal{L}_k} B_{lh} b^O_{ol}) = N^O & \forall h \in \mathcal{H} & (5)^{\text{T3}} \\
& x_{hio} + \sum_{c\in\mathcal{C}} s^I_{hic} \leq F \sum_{k\in\mathcal{K}_i} W^I_{ikh} w^I_{ik} & \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O} & (6) \\
& x_{hio} + s^O_{ho} \leq F \sum_{k\in\mathcal{K}_o} W^O_{okh} w^O_{ok} & \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O} & (7) \\
& \sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co} x_{hio} + \sum_{h\in\mathcal{H}} s^I_{hic} = Q_{ic} & \forall i \in \mathcal{I}, c \in \mathcal{C} & (8) \\
& \sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s^O_{ho} = F & \forall o \in \mathcal{O} & (9) \\
& \sum_{o\in O} x_{hio} + \sum_{d\in D} s^I_{hid} \leq M & \forall i \in \mathcal{I}, h \in \mathcal{H} & (10) \\
& \sum_{k\in\mathcal{K}_i} w^I_{ik} = 1 & \forall i \in \mathcal{I} & (11) \\
& \sum_{k\in\mathcal{K}_o} w^O_{ok} = 1 & \forall o \in \mathcal{O} & (12) \\
& s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s^I_{hic} - \sum_{o\in\mathcal{O}} Z_{co} s^O_{ho} & \forall c \in \mathcal{C}, h \in \mathcal{H} \setminus \{0\} & (13) \\
& s_{0c} = \sum_{i\in\mathcal{I}} s^I_{0ic} - \sum_{o\in\mathcal{O}} Z_{co} s^O_{0o} & \forall c \in \mathcal{C} & (14) \\
& \sum_{l\in\mathcal{L}} b^I_{il} = 1 & \forall i \in \mathcal{I} & (15)^{\text{T3}} \\
& \sum_{l\in\mathcal{L}} b^O_{ol} = 1 & \forall o \in \mathcal{O} & (16)^{\text{T3}} \\
& avg = \frac{1}{|\mathcal{I}|+|\mathcal{O}|} \left(\sum_{i\in\mathcal{I},l\in\mathcal{L}} D_l b^I_{il} + \sum_{o\in\mathcal{O},l\in\mathcal{L}} D_l b^O_{ol}\right) & & (17)^{\text{T3}} \\
& & & \\
& x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc}, n^I, n^O \in \mathbb{N}^+ & \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O}, c \in \mathcal{C} & \\
& w^I_{ik}, w^O_{ok}, b^I_{il}, b^O_{ol} \in \{0,1\} & \forall i \in \mathcal{I}, o \in \mathcal{O}, k \in \mathcal{K}, l \in \mathcal{L} &
\end{array}
$$

(IP*)$^{\text{T3}}$

4.2.3.4 *Model* T4

T4 uses the same model as T3, but with a different objective function. The idea of the method comes from Hazır *et al.* [101], who show that the maximum weighted slack of a project is correlated with the punctuality of the project (roughly, the probability that it ends before its deadline). They define the weight of each task as its number of successors in the Gantt chart of the project. The idea is adapted to the cross-docking context as shown in algorithm 4.3.

1. Run IP* for small instances, or H2 for larger instances.
2. Following a FIFO policy, match each inbound truck to an inbound door and each outbound truck to an outbound door.
3. Calculate the number of successors σ_i^I of each inbound truck $i \in \mathcal{I}$, and the number of successors σ_o^O of each outbound truck $o \in \mathcal{O}$.
4. Run $(\text{IP}^*)^{\text{T4}}$, *i. e.* $(\text{IP}^*)^{\text{T3}}$ with the following objective function:

$$\max \sum_{i \in \mathcal{I}, l \in \mathcal{L}} \sigma_i^I b_{il}^I + \sum_{o \in \mathcal{O}, l \in \mathcal{L}} \sigma_o^O b_{ol}^O$$

Algorithm 4.3: Maximizing the weighted sum of buffers

4.3 NUMERICAL RESULTS

In this section, the different models described previously are tested in order to compare their performances in terms of robustness.

4.3.1 *Methodology*

The instance sets tested in this chapter are the ones described in section 2.2.3.

The robustness of the schedules generated by the different models is assessed following the methodology detailed in chapter 3. Particularly, the robustness is measured using the indicators introduced in section 3.4:

– R^{transfer} (Equation 3.1 on page 86),
– R^{unload} (Equation 3.2 on page 87),
– R^{arrival} (Equation 3.3 on page 87).

4.3.2 *Results on instance* `set3+3`

For each instance of `set3+3`, a truck schedule is calculated with IP* and with the different models detailed in section 4.2. Figure 4.3 shows the relative value of the robustness indicator (average on all the instances of the set) compared to IP*, for each of the robust versions proposed. When the value is positive for a source of uncertainty (transfer time, unloading time or truck arrival time), it means that the robustness of the schedule regarding this source of uncertainty is better than the robustness of a schedule generated with IP*. A negative value means the robustness is degraded compared to IP*. Figure 4.4 shows the standard deviations for the values averaged in Figure 4.3. Because robustness can be created at the expense of stock level, Figure 4.5 monitors in each model the average increase of the number of pallets stored temporarily.

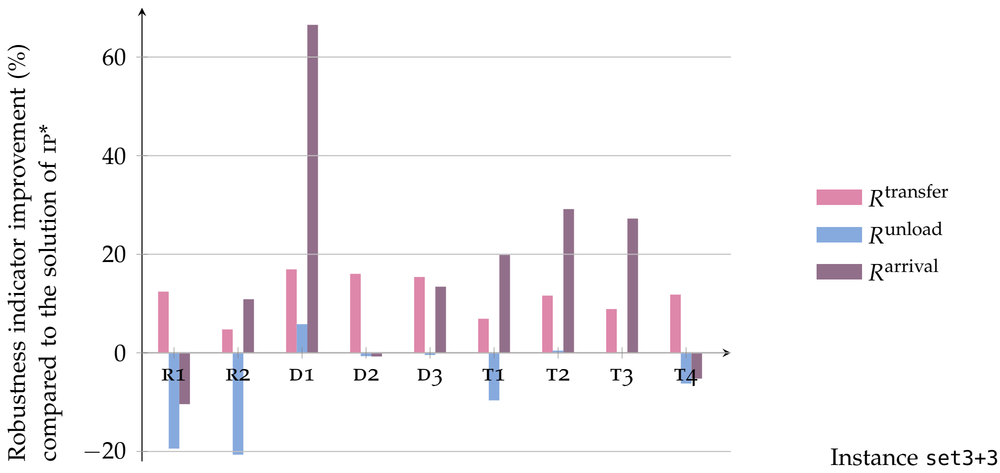


Figure 4.3: Robustness evaluation

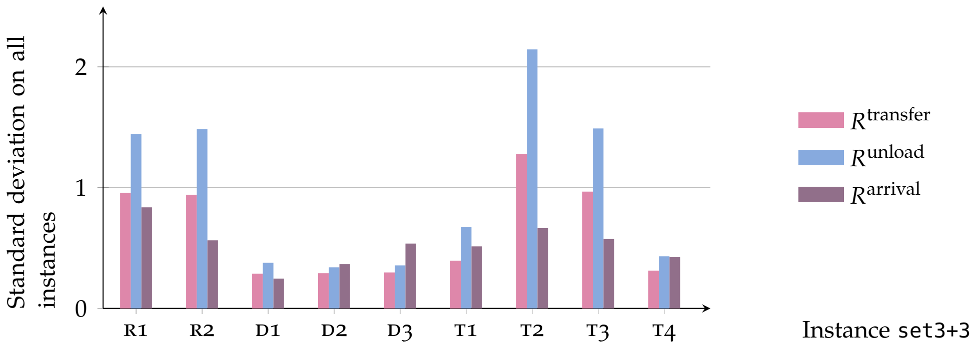


Figure 4.4: Standard deviation of the results

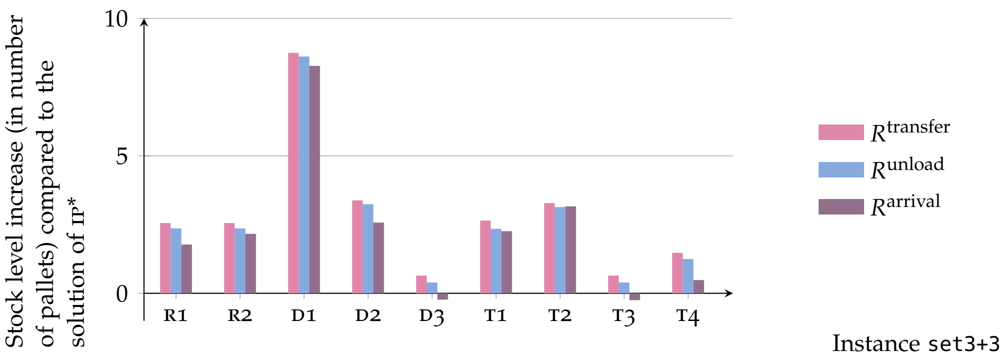


Figure 4.5: Stock level increase for the different models

robust optimization techniques Model r1 follows a minimax logic, *e. g.* finds a schedule which gives the best results in the worst scenario. However, "best" in this case is measured through the value of the objective function (truck presence time window penalties, number of pallets in storage) and not in terms of robustness. Because only the worst case is taken into account, the average robustness of the schedules obtained is not very good, except regarding robustness to changes in the transfer time.

Model r2 improves the average result regarding robustness to the truck arrival time – which could be expected since the objective is to have a performance when trucks are delayed as close as possible to the performance in the deterministic case. Yet the robustness against changes in the truck arrival times is not very high compared to the results of other methods, and compensated by a bad resistance to changes in unloading time.

Both models create a moderate increase in the level of temporary storage.

resource redundancy The three models d1, d2 and d3 have a positive or null robustness improvement compared to ip* for the three sources of uncertainty, while having a low standard deviation which means that this result is homogeneous on the different instances. Method d1, which minimizes the average number of trucks present at the doors, has an excellent robustness when facing changes in the truck arrival times: 67% improvement compared to the robustness of ip*. The price for robustness is paid here by an important increase of the amount of pallet stored. d3 offers a good compromise between robustness and stock level.

d1 *is very robust but increases storage.*

d3 *offers a good robustness/storage trade-off.*

time redundancy The time redundancy-based models that give the best results (positive or null improvement in robustness for all three sources of variability) are t2 and t3. Besides, t3 has almost no impact on the number of pallets put in storage – on the average it is even a bit smaller than ip* when truck arrival times are variable. However, t2 and t3 also have the largest standard deviations, which means that the quality of the results can be quite different depending on the instance. It is interesting to note that t1 and t2, which use a rather simple post-treatment (linear in function of the number of trucks), give better results than some IP formulations, with only a moderate increase in the amount of pallet stored.

t2 *is a linear post-treatment that gives good results.*

4.4 conclusion

This chapter proposes various reformulations of the crossdock truck scheduling problem introduced in chapter 2, with the aim of improving the robustness of the schedules obtained. The robust versions proposed make use of standard robust optimization techniques, but also techniques inspired from robust project scheduling methods: resource redundancy (*i.e.* door redundancy when applied to cross-docks) and time redundancy (*i.e.* insertion of buffers or slacks). Nine different versions are numerically tested using the simulation model, methodology and robustness indicators introduced in chapter 3. The comparison shows that the methods based on resource redundancy, a method barely used in project scheduling because of its expensive cost, give the best results overall in the cross-docking case. Minimizing the average number of trucks docked at a given door is a good way to ensure robustness in the schedule, but increases storage.

The proposed models could of course be improved. Models R1 and R2 that use robust optimization techniques could include more scenarios in order to take more potential cases into account. The models based on time-redundancy could be refined by making the length of the buffer a function of other parameters such as the door utilization, the transfer time, unloading time... The simulation model could also be used to find the best buffer length.

Developing a simulation-optimization approach would be another way to improve the robustness of the models. In models R1 and R2, the tabu search is led by the value of the objective function Π_0. By connecting the tabu search to the simulation model, the value of the robustness indicator(s) could be used instead, to lead the search for robust schedules.

Finally, let us remember how robustness is achieved in practice in a crossdock: unexpected changes in the schedule will be met by a higher engagement of the workers in order to finish the work. When a critical truck is late, it is compensated internally by putting a high priority or allocating more resources for unloading/loading this truck. The allocation of resources is thus a major issue to consider: it is addressed in chapter 5.

Allez, le temps est cher: il le faut employer.

— Jean Racine

Chapter 5

OPTIMIZING CROSSDOCK EMPLOYEE SCHEDULING

The study carried out in chapter 1 shows that human and material resources of the platform are often assumed infinite in the literature, whereas platform managers find it crucial to match the resources to the activity volume. Constraints for scheduling and rostering are numerous: logistic employees are multi-skilled employees and have flexible working hours or short-term contracts. Legal constraints and handling equipments' capacities should also be met. This chapter describes a model supporting the chain of decisions from weekly timetabling to daily rostering (detailed task allocation). The problem is divided into three sub-problems depending on the type of decision to be made: workforce dimensioning, task allocation for a week, and detailed rostering for a day. The three decisions are made sequentially; each step is modeled as a Mixed and Integer Linear Program. The proposed models are tested with industrial data as well as generated instances.

The work presented in this chapter is also presented in the following article:

Ladier, A.-L., Alpan, G., and Penz, B. 2014. Joint employee weekly timetabling and daily rostering: A decision-support tool for a logistics platform. *European Journal of Operational Research* 234, 1, 278-291.

OPTIMISATION DES EMPLOIS DU TEMPS

L'étude du chapitre 1 montre que les ressources humaines et matérielles à l'intérieur de la plateforme sont souvent supposées infinies dans la littérature, alors que pour les managers de plateforme l'adéquation des ressources au volume d'activité est cruciale en termes de performance. Les emplois du temps doivent respecter de nombreuses contraintes :

– les opérateurs sont polyvalents, avec un profil de compétences spécifique pour chacun ;
– la modulation est autorisée (en France par exemple, selon les accords d'entreprise, les 35 heures par semaine peuvent être réalisées en moyenne sur l'année) ;
– l'embauche d'intérimaires est possible, avec des coûts qui dépendent des compétences ;
– le nombre d'engins de manutention disponibles, la pénibilité des tâches, l'équité et la régularité du planning obtenu. . . doivent être également pris en compte.

Ce chapitre présente un modèle permettant d'accompagner la chaîne de décision qui va de la réalisation du planning hebdomadaire à l'allocation quotidienne des tâches. Le problème est divisé en trois sous-problèmes en fonction du niveau de décision : dimensionnement de l'équipe, allocation des tâches pour la semaine, et planning détaillé de la journée. Ces étapes sont modélisées par trois programmes linéaires mixtes résolus de façon séquentielle. Ils permettent d'affecter aux employés leur volume de travail par jour (MILP1), leurs horaires exacts et leurs tâches avec une précision à l'heure (MILP2), et leurs tâches pour un jour donné avec une précision au quart d'heure (MILP3). Les modèles proposés sont testés sur des données industrielles et des instances générées aléatoirement. Les observations menées dans un contexte industriel permettent de montrer en quoi le modèle est un outil d'aide à la décision pour les managers. L'outil est actuellement utilisé par l'entreprise qui a fourni les données industrielles. Les résultats sur les instances générées permettent de déterminer sous quelles conditions les modèles peuvent être résolus en un temps raisonnable. Une étude de sensibilité est également menée pour observer les effets d'un changements sur les données d'entrée entre l'exécution de MILP2 et celle de MILP3.

5 OPTIMIZING CROSSDOCK EMPLOYEE SCHEDULING

In chapter 1, the comparative study of the cross-docking literature and the practice of crossdock managers has shown that workforce management is a problem of crucial importance for the managers which is barely addressed in the cross-docking literature. Most articles consider the human resources within the platform as unlimited. In chapter 2, chapter 3 and chapter 4, the platform capacity is assumed to be fixed and is equal to M during the entire planning horizon. In order to make this assumption more realistic, it is necessary to know exactly how many workers are present in the platform and available for the different operations to be carried out (*e.g.* unloading, control, transfer…). Therefore, this chapter studies a personnel scheduling problem in the context of a logistic platform. Note that it can apply to any type of logistic operations and not only cross-docking.

5.1 EMPLOYEE TIMETABLING AND ROSTERING FOR LOGISTICS: PROBLEM DESCRIPTION

As noted in section 1.2.1.1, goods can be moved inside the cross-docking platform either manually, with an automated system (*e.g.* conveyor belts) or with a combination of both. Automation can also be used for storage (automated storage and retrieval systems) and picking (pick-to-light systems) – see *e.g.* Baker and Halim [15] or Granlund [87]. Note that these systems support human's work but do not replace it. In general, automated systems represent heavy investments, but are feared to be not flexible enough to meet changing market requirements (Baker and Halim [15]). Therefore, automation is generally adopted by companies dealing with a limited range of product types, in a stable or growing market (*e.g.* postal and parcel services). For logistic service providers, whose survival depends on their flexibility, the operations stay mainly manual. Manpower is therefore the first cost center in logistics and especially for logistics providers (see Graham [86] and van den Berg [203]).

It is thus crucial to stick to the activity volume when dimensioning the task force. A difficulty is that the workload is variable: the number of arriving trucks and the number of orders to be prepared change every day. For instance, one third of the warehouses in France have a seasonal activity (Service de l'observation et des Statistiques [177]).

In a recent article, de Leeuw and Wiers [60] study the effects of the financial crisis over warehouse manpower planning strategies. They show that in times of financial crisis, companies in the Netherlands increase the number of temporary workers, increase the use of flexible planning for employees with fixed contracts, and increase the workload balancing. Techniques for workload balancing can include planning time slots for incoming trucks (as done in chapter 2) or postponing some tasks to the next day if feasible. They also show that flexible planning has a strong positive influence on warehouse performance.

How to build a flexible planning? The number of working hours for a given employee may differ from one week to another, and short-term contracts are also used to ensure more flexibility – 80% of the French warehouses use temporary workers according to the statistics department of the French ministry for sustainable development (Service de l'observation et des Statistiques [177]). These parameters, together with other constraints such as the employees' qualifications, vacations, the handling equipment availability, *etc.*, make *weekly timetabling* and *daily rostering* a complex process.

See definitions of weekly timetabling and daily rostering in section 5.1.2.

Although weekly timetabling and daily rostering are intertwined, they are often treated separately in the literature; we propose to deal with the two of them together through sequential solving. Logistics is not a common application area for personnel scheduling problems, and the few existing papers use heuristic methods to solve the problem. The decision-support tool proposed in this chapter meets the specific requirements of a logistics platform to support the personnel scheduling process for warehousing operations, and its solution is based on optimal methods, *i.e.* Mixed and Integer Linear Programs (MILPs). The problem is divided into three steps, each representing a decision to be made. Each step is modeled by a Mixed and Integer Linear Program.

The assumptions for this problem are detailed in section 5.1.1, followed by a literature review of timetabling and rostering problems (section 5.1.2). An overview of the model is given in section 5.2: the first part of the model, namely the weekly timetabling (step 1 and step 2) is detailed in section 5.2.1, while section 5.2.2 deals with the detailed daily rostering (step 3). An analysis of the complexity of the different steps is given in section 5.2.3. Section 5.3 presents the numerical results, and concluding remarks are given in section 5.4.

5.1.1 *Assumptions*

The goal is to define a model which can be used in logistics platforms to generate personnel schedules based on optimal methods. To be as close as possible to an industrial context when building the model and defining its main assumptions, we observed the scheduling process within a warehouse where the timetabling generation is done manually.

According to the agreements signed with the trade unions, the working hours for the following week have to be communicated to the employees seven days in advance. The daily roster, however, can be given every morning or even redefined at any time during the day. Of course the working hours of each employee in this detailed roster must be as close as possible to what has been announced a week before.

The workload is varying over time, while the employees' working hours are flexible. Two types of employees are considered: regular employees and temporary workers. For regular employees, various shifts are possible as long as they respect the trade agreements. Temporary workers with short-time contracts, though, do not have flexible working times: they are hired for the exact number of hours allowed by the law per week. All employees (regular employees and temporary workers) have different qualifications for each task, depending on their training. Of course, legal requirements and safety principles should also be met in the model.

The problem presented in this chapter falls in the category of "multi-day personnel scheduling problems" defined by Brucker *et al.* [36] in their general model for personnel scheduling.

5.1.2 *Similar problems in the literature*

Following Ernst *et al.* [67], we use the words *personnel scheduling* to describe the whole process of constructing work timetables for an organization's staff, in order to satisfy the demand for its goods or services. As mentioned by Musliu *et al.* [152], personnel scheduling algorithms consist of different stages related to each other, that can be solved simultaneously or in sequence, depending on the context.

Brucker *et al.* [36] underline that personnel scheduling problems can be decomposed into two levels: in the first stage, the working days are assigned to the employees, whereas the second stage assigns a shift for each employee working on a given day, and a task for which the employee is qualified on each working period. In this document, we call *weekly timetabling* the first stage of the process which consists of determining the number of employees needed and allocating these employees to shifts (sets of consecutive time periods within a day) in order to meet the forecast workload. The second stage of the process matches Wren's definition of *rostering* as:

> "the placing, subject to constraints, of resources into slots in a pattern. One may seek to minimize some objective, or simply to obtain a feasible allocation. Often the resources will rotate through a roster".
>
> Wren [215]

In this document, the expression *daily rostering* therefore refers to the assignment of tasks to employees on a daily level.

Our literature review will focus on two aspects: firstly, section 5.1.2.1 focuses on the application areas of personal scheduling problems, to see how the logistics field relates with the fields covered by current research. Secondly, in section 5.1.2.2 we have a closer look at the methods used in the literature to solve weekly timetabling and daily rostering problems.

5.1.2.1 *Personnel scheduling in logistics*

The logistics industry faces several challenges which are specific to this field:

- The highly variable demand makes the workload very different from one day to another, which means that regular patterns cannot be used to create the workers' timetables;
- The qualifications are very specific to a person: two employees are very likely to have different skills and different licenses to drive the handling equipment. Therefore, the set of tasks mastered by a given employee will be different from the set of tasks mastered by any of his colleagues, and clustering the employees according to their skills does not simplify the problem – see De Bruecker *et al.* [58] for a detailed review of workforce planning problems incorporating skills and an analysis of the impact of different skill types on the problem formulation;
- The unequal distribution of busy periods over a day does not fit a standard 8-hour shift: supervisors must therefore assign shorter or longer shifts, force some employees to take a day off, or hire temporary workers.

Crew scheduling and nurse rostering are active streams of research in personnel scheduling.

Personnel scheduling questions have been broadly studied for transportation systems (including airlines, railways and buses): the constraints tackled by the so-called *crew scheduling* problems are very specific, since the location of the crews is also a variable. The interested reader can refer to Castillo-Salazar *et al.* [41] for a survey on workforce scheduling and routing. *Nurse scheduling* and, more generally, health care systems scheduling is also a major application area (see the survey by Burke *et al.* [37]), in which the problems are highly constrained because hospitals work around the clock. The main differences between the health care field and logistics are:

- The relative simplicity of the qualifications profiles used in nurse scheduling. As mentioned earlier, a logistics employee has qualifications that allow him to work only on specific tasks, while a nurse has one qualification which allows her to do all the tasks. Therefore, the daily rostering is not needed for nurses, since they know precisely what they are supposed to do when assigned to a given shift. The problem can be solved on a shift level.

– The shape of the coverage function (number of employees required each hour). As highlighted by De Causmaecker *et al.* [59], hospital personnel scheduling problems are *permanence centered*, while warehouse personnel planning are based on *fluctuating demand*.

Overall, the granularity of the nurse timetabling problems is larger than staff timetabling for logistics and, more generally, for the service industry.

The service industries whose characteristics and requirements are the closest to the logistics area are retailing, call centers and postal service; for instance, the model proposed by Bard *et al.* [17] to schedule the United States Postal Service staff meets most of the constraints encountered in logistics operations. However, they focus on the long-range planning problem rather than the weekly scheduling problem. The weekly personnel scheduling problems raised in the US Postal Service mail processing are addressed by Wan [211], who also deals with the US Postal Service distribution centers, whose activities are typical logistics operations. But like Bard *et al.* [17], he considers a homogeneous workforce, without distinctions in skills and qualifications.

The literature studying warehouse personnel scheduling as such is still very limited: no paper appears in the comprehensive review made by Ernst *et al.* [66], covering the literature until 2004 of more than 700 analyzed sources dealing with personnel scheduling problems. The review by De Bruecker *et al.* [58], covering the articles published between 2004 and 2012 that incorporate skills in the timetabling problem, does not include any article regarding warehouse personnel scheduling either. Only De Causmaecker *et al.* [59] mention this field as an application area, since a small warehouse (20 employees) was included in the sample of Belgian companies they investigated to classify the scheduling problems. A recent state-of-the-art by van den Bergh *et al.* [204] reviews 291 articles from 2004 to 2012, in which Günther and Nissen [94, 95] are the only ones dealing with a real-world scheduling problem in logistics, comparing three heuristics and an evolutionary method to solve a daily rostering problem for a German logistics service provider with 65 employees. The model proposed by these authors is a multi-objective model. They seek to minimize the over and under-staffing, the extra hours worked every week, and the cases where the working days are too short, too long, or split up during a working day. The industrial data used is in open access. We will come back to these data at the numerical experiments section (section 5.3.2).

Very few articles deal with personnel scheduling for logistic plaforms.

5.1.2.2 *Joint approaches for weekly timetabling and daily rostering*

In this chapter, we propose to solve in sequence a weekly timetabling and a daily rostering problem.

From the articles gathered by Ernst *et al.* [66], it can be noticed that these concepts (named a bit differently in the review, since the authors use the words "workforce planning", "shift scheduling" and "task assignment") are never studied at the same time: amongst the articles reviewed, 163 deal with workforce planning and shift scheduling, 33 for task assignment, but none tackles these problems together.

Since the review by Ernst *et al.* [66] was conducted in 2004, two articles proposed some global models to solve a timetabling problem: Detienne *et al.* [61] and Naudin *et al.* [154]. The similarity with our approach resides in the fact that they also consider the overall problem as a two-stage decision problem (a weekly stage and a daily stage). In their case, this idea is exploited to propose bounds or decomposition methods that can help solving one overall model. Detienne *et al.* [61] use this idea to implement a Lagrangian lower bound for their model. They also propose a multi-dimensional multi-choice knapsack problem which aggregates the two decision stages in one, but the latter formulation generates an exponential number of constraints. Naudin *et al.* [154] propose two decompositions: a Dantzig-Wolfe decomposition reformulated with mid-term variables, and another one with long-term variables. These two approaches are not easily applicable in our case because of the multiple constraints we have. In the current chapter, we use the decomposition idea in order to propose a sequential approach where the problem is divided into steps solved one after the other, each stage using as an input the output of the previous stage. Each phase having only limited information about the others, the solution is unlikely to be optimal, but this approach can solve large timetabling problems. Another advantage of our model compared to Detienne *et al.* [61] and Naudin *et al.* [154] is that the outcome of our approach is not only a daily timetable but both the weekly schedule and the daily timetable. From a managerial point of view, both are of importance. The managers need the weekly schedule for workforce dimensioning and planning, and the daily rostering for operations management. Furthermore, having both gives a certain flexibility in case of unexpected events. For instance, the daily rostering can be readjusted very quickly based on a new piece of information, which was not available when the weekly schedule was done.

A sequential approach with two stages is used in a recent paper by van Veldhoven *et al.* [205] but to solve a different problem: the nurse days-off scheduling problem. The first stage specifies the days off for each employee (days off scheduling), then the second phase specifies which shifts are actually assigned to the employees on their working days. Each phase is solved with an integer program. We note that our problem is of finer granularity, since we deal with the daily rostering as well.

5.1.2.3 *Similar approaches to each step of of the sequential approach*

Each stage of our model, taken separately, presents some similarities with problems which have been modeled already:

WORKFORCE DIMENSIONING. This step is close to the one solved by Eitzen *et al.* [65] for an Australian power station: the employees have different skill qualifications and need to be allocated under legal constraints, while ensuring the equity of the outcome. The authors formulated the problem as a generalized set-covering problem minimizing the total under-staffing, and tested various solution strategies. Only the method of branch and price is capable of finding a provably optimal solution for a problem size of 20 to 110 employees. Note also that Eitzen *et al.* [65] do not consider hiring temporary workers if the demand gets too high. The review by van den Bergh *et al.* [204] shows that the possibility of hiring interim or casual workers is not very common in the personnel scheduling literature.

ASSIGNING SHIFTS AND TASKS TO EMPLOYEES. This second step is close to the one described by Schaerf and Meisels [176], although they do not give the exact formulation of their model in this paper. Their generalized local search is tested for nurses in a hospital department and for a production line in a factory, with 20 to 50 employees, 100 to 300 tasks and 20 to 40 shifts, after relaxing all soft constraints. Their coverage function is less precise than the one we use, since it only gives a number of employees that should be present during each shift. Also, it is not clear whether the length of the shifts can vary in their case. Another model close to ours is the one proposed by Dahmen and Rekik [53], who deal with a multi-activity shift problem. The goal is to construct the shifts and assign the activities for employees with various qualifications (although all the employees are qualified for all tasks in the instances used to test the model). The constraints considered are very similar to ours: over and under-staffing are penalized, and the tasks have minimum and maximum durations. The limit of this model is that it supposes an explicit enumeration of all admissible shifts with assigned activities for each employee. This, in practice, is quite complex to do when the number of activities or employees is high. The authors propose a hybrid heuristic to solve the problem, combining tabu search and branch and bound.

DAILY ROSTERING. The daily rostering problem has been modeled by Campbell and Diaby [38] in the case of multi-skilled workers. In their model, a worker less skilled than another needs more time to complete his work. They propose a linear program for the special case of binary capabilities, and an assignment heuristic for the general allocation problem. More recently, other problems closer to our third stage have been studied by Smet and Vanden Berghe [184] and

Lequy *et al*. [118, 119]. Smet and Vanden Berghe [184] deal with a shift minimization personnel task scheduling problem, where the objective is to assign tasks to multi-skilled employees (with binary capabilities) while minimizing the number of employees used. Solutions are obtained with a very large-scale neighborhood search algorithm, combining metaheuristics and exact approaches. A model closer to our approach is the multi-activity assignment problem proposed by Lequy *et al.* [118]. The work shifts being already assigned to the employees, the problem is to assign activities, taking the qualifications into account and covering the demand as much as possible on the planning horizon (from one day to one week). The objective is to minimize the under-staffing, over-staffing and transition costs (paid when an employee changes activity). This work is extended by Lequy *et al.* [119]: the workload is now divided between tasks which are uninterruptible pieces of work, and activities for which preemption is allowed. The solution strategy proposed is a two-stage heuristic: the task assignment is done first, then the activities are assigned considering the fixed tasks.

It is important to note that in all those papers, the employees' shifts cannot be changed anymore at this stage. In our model however, shift changes are allowed in exchange for a penalty cost, if there are differences between the forecast and the actual workload occurring on that day.

The articles cited above solve a part of the global problem we want to solve, and they mainly do it through heuristics or metaheuristics. The originality of our approach is to combine the workforce scheduling problem and the daily rostering through sequential solving, each step being modeled by a MILP solved to optimum.

5.1.3 *Input data: notations*

In the following sections, the input data and the decision variables are defined over these sets:

$\mathcal{E}$ set of employees considered in the timetabling operation.

$\mathcal{E}^{fixed}$ subset of $\mathcal{E}$, set of employees whose shift is fixed beforehand. These employees are working under a special contract (pre-retirement, for instance) and their working time is fixed instead of flexible. Their exact tasks during that time still have to be calculated.

$\mathcal{T}$ set of tasks to be processed by the employees. Two tasks are different if they require different abilities or different handling machines. The time when the tasks have to be done depends on their nature: some tasks must be carried out in precise time windows, while some can be carried out at any time during the day. We therefore split the tasks into two groups:

- $\mathcal{T}^1$ subset of $\mathcal{T}$, set of tasks whose workload is defined precisely, hour per hour. For instance, containers have to be unloaded right after their arrival, so the workload for the task "unloading containers" is defined hour per hour. Note that this definition helps expressing precedence relationships among tasks in the data set. For each arriving container, the distribution of the workload over all tasks are estimated. For example, if a container arrives at 8AM, a non-zero workload will be estimated for the "unloading" task at 8AM, then the workload for the "scanning and computer reception" task will be estimated at 9AM. The constraints on consecutive tasks are therefore not needed in the model.
- $\mathcal{T}^2$ subset of $\mathcal{T}$, complementary to $\mathcal{T}^1$, set of tasks whose workload is defined for a whole slot. For instance, stocktaking is a task that can be completed at any time during the day, the workload for this task is therefore defined for the whole slot 8AM–5PM.

$\mathcal{P}$ set of temporary workers profiles. In case the workload is too heavy compared to the workforce available, the decision-support tool will suggest to hire temporary workers of a given profile. A temporary worker profile is a set of tasks that this type of worker can handle. For example, the profile of an "order picker" could be {manual unloading, picking, wrapping}.

$\mathcal{D}$ set of working days considered for the weekly schedule. That can be five to seven, depending on whether work over the weekend is allowed or not.

$\mathcal{H}$ set of working hours in a day. That can be eight to twenty-four, depending on whether the activity runs with one, two or three shifts a day.

$\mathcal{S}$ set of possible shifts. A shift, for example, is "8AM–4PM" or "10AM–6PM". Two shifts are different if they have different beginning and/or ending times.

$\mathcal{Q}$ set of intervals considered for the daily rostering. The unit may be smaller than an hour, *e.g.* a quarter of an hour.

5.2 SEQUENTIAL SOLVING

Since there are two different time scales in the decisions to be made, the problem can be split into two distinct phases. First, working days and shifts are assigned to employees for one week (*weekly timetabling*); then the weekly timetable is used as a basis to re-assign tasks within a day with more precision, taking into account the possibly new data which may arrive in the meantime (*daily rostering*).

Looking more closely at the *weekly timetabling* problem, we can see that it is also a two-stage decision. First, the workforce has to be dimensioned (decision about the number of employees to hire on short-

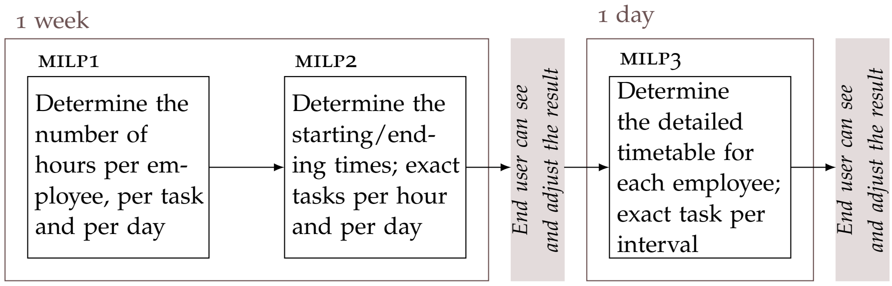


Figure 5.1: Overview of the decision-support tool

term contracts, and the total number of hours worked) before deciding upon the shifts themselves (when each employee should start and finish his day) and the task allocation. The scheduling system is therefore composed of three different MILPs, as shown in Figure 5.1. These are referred to in this chapter as MILP1, MILP2 and MILP3. The *weekly timetabling* part, composed of MILP1 and MILP2, is detailed in section 5.2.1, and the *daily rostering* part (MILP3) in section 5.2.2. What needs to be noted at this point is the fact that some inputs of MILP2 and MILP3 are the outputs of MILP1 and MILP2 respectively.

The objective of each MILP is to get a feasible solution under the hard constraints related to legal requirements, while minimizing unwanted situations like non-equity, over-staffing or having an employee working on the same task for too long. A penalty point is counted for the occurrence of each one of these situations, and the objective function is a weighted sum of the penalty points.

5.2.1 *Weekly timetabling*

This section presents in detail how to obtain a weekly timetable for the employees. First, the workforce dimensioning is calculated using MILP1, and this data is used to process the weekly shift allocation in MILP2.

5.2.1.1 MILP1

The aim of the first Mixed and Integer Linear Program is to define the workload per person and per day. The decisions made are therefore about the number of temporary workers to hire, and the number of hours per task assigned to a worker (temporary or regular) each day, for a one-week horizon.

Initially, a pre-treatment of the input data is made, in order to reduce the size of the linear programs in terms of number of constraints. We have information about:

– the abilities ($\mathcal{E} \times \mathcal{T}$): for each task, whether or not each employee is able to carry it out;

– the availabilities ($\mathcal{E} \times \mathcal{D}$): for each day, whether or not each employee is present;
– the work planning ($\mathcal{D} \times \mathcal{T}$): for each day, whether or not each task needs to be carried out on that day.

Instead of considering these three bi-dimensional matrices in the linear program, we build a three-dimensional binary matrix that puts this information together without redundancies. We call this matrix X and define it over $\mathcal{E} \times \mathcal{D} \times \mathcal{T}$. $X_{edt} = 0$ if employee e is not qualified for task t, or if employee e is not available on day d, or if there is no work needed for task t on day d; $X_{edt} = 1$ otherwise.

INPUT DATA. The following data are used as inputs:

X_{edt} Data matrix as defined above.

W_{td} Workload (in working hours) for task $t \in \mathcal{T}$ and day $d \in \mathcal{D}$.

N_{td} Minimum number of people needed at the same time to carry out task $t \in \mathcal{T}$ on day $d \in \mathcal{D}$.

Q_{et} Non-binary qualifications of employee $e \in \mathcal{E}$ for task $t \in \mathcal{T}$, defined on $\{0..\zeta\}$ where $\zeta \in \mathbb{N}^*$. The value of Q_{et} depends on the level of experience of the employee e for a given task t.

P_{pt} Temporary workers profile description: $P_{pt} = 1$ if a worker with profile $p \in \mathcal{P}$ is qualified for task $t \in \mathcal{T}$, 0 otherwise.

C_p Temporary workers cost C_p is the cost of hiring a worker with profile $p \in \mathcal{P}$.

Max_t Maximum amount of time (in hours) that a worker can spend per day on task $t \in \mathcal{T}$. This value enables one to respect safety and ergonomics principles.

F_{ed} Working time of the employee $e \in \mathcal{E}^{fixed}$, whose shift is defined beforehand, on day $d \in \mathcal{D}$.

In order to be as general as possible, we note min^{day} the minimum number of hours that an employee can work per day, max^{day} the maximum number of daily hours, and max^{week} the maximum number of hours per week permitted by the law. The model can therefore be adapted to different labor legislations or local agreements.

In France, $min^{day} = 4$, $max^{day} = 10$ and $max^{week} = 44$.

DECISION VARIABLES. This step uses the following decision variables regarding the regular workers:

h_{edt} Number of hours worked by employee $e \in \mathcal{E}$ on day $d \in \mathcal{D}$ and task $t \in \mathcal{T}$.

p_{ed} Presence of employee $e \in \mathcal{E}$ on day $d \in \mathcal{D}$: $p_{ed} = 1$ if e works on day d, 0 otherwise.

x_{edt} Task allocation: $x_{edt} = 1$ if employee $e \in \mathcal{E}$ works on task $t \in \mathcal{T}$ on day $d \in \mathcal{D}$, 0 otherwise.

Regarding the temporary workers:

h_{dtp} Number of hours worked by all temporary workers of profile $p \in \mathcal{P}$ on day $d \in \mathcal{D}$ and task $t \in \mathcal{T}$.

n_p Number of temporary workers hired with profile $p \in \mathcal{P}$.

OBJECTIVE FUNCTION. The objective function is a weighted sum of the penalties listed below.

Π_1^α *Temporary workers penalty*. We give C_p penalty points for each temporary worker of profile $p \in \mathcal{P}$ we suggest to hire.

Π_1^β *Qualifications penalty*. We give $(\zeta - \lambda)$ penalty points for each hour spent on a task by an employee who has a qualification λ.

Π_1^γ *Equity penalty*. We give a penalty point for each hour of difference between two employees' total numbers of working hours on one day.

Π_1^δ *Ergonomic penalty*. We give a penalty point for each hour in excess compared to the maximum amount of time allowable for a task, per worker and per day.

Π_1^ε *Unplanned absence penalties*. We give a penalty point each time we force a regular employee to take a day off, which was not planned by the employee himself (*i. e.* not defined in the matrix X). From the company's point of view, these extra days off are a good way to compensate overtime work.

MODEL. MILP1 is formulated as shown below. Note that some constraints use absolute values and are therefore not linear; but they can be easily linearized as described in Equation 4.4 on page 107.

The objective function minimizes the weighted sum of all penalties, defined by constraints (24) to (28) as detailed in the penalty list

$$
\begin{array}{lll r}
\min & \alpha_1\Pi_1^\alpha + \beta_1\Pi_1^\beta + \gamma_1\Pi_1^\gamma + \delta_1\Pi_1^\delta + \varepsilon_1\Pi_1^\varepsilon & & \\
\text{s.t.} & \Pi_1^\alpha = \sum_{p\in\mathcal{P}} n_p C_p & & (24) \\
& \Pi_1^\beta = \sum_{e\in\mathcal{E},d\in\mathcal{D},t\in\mathcal{T}} (\zeta - Q_{et}x_{edt}) & & (25) \\
& \Pi_1^\gamma = \sum_{e_1,e_2\in\mathcal{E},d\in\mathcal{D}} \left|\sum_{t\in\mathcal{T}} h_{e_1dt} - \sum_{t\in\mathcal{T}} h_{e_2dt}\right| & & (26) \\
& \Pi_1^\delta = \sum_{e\in\mathcal{E},d\in\mathcal{D},t\in\mathcal{T}} (h_{edt} - Max_t) & & (27) \\
& \Pi_1^\varepsilon = \sum_{e\in\mathcal{E},d\in\mathcal{D}} (\sum_{t\in\mathcal{T}} X_{edt} - p_{ed}) & & (28) \\
& min^{\text{day}} p_{ed} \le \sum_{t\in\mathcal{T}} h_{edt} \le max^{\text{day}} p_{ed} & \forall e\in\mathcal{E}, d\in\mathcal{D} & (29) \\
& \sum_{d\in\mathcal{D},t\in\mathcal{T}} h_{edt} \le max^{\text{week}} & \forall e\in\mathcal{E} & (30) \\
& \sum_{t\in\mathcal{T}} h_{dtp} = 7n_p & \forall p\in\mathcal{P}, d\in\mathcal{D} & (31) \\
& \sum_{t\in\mathcal{T}} h_{edt} = p_{ed}F_{ed} & \forall e\in\mathcal{E}^{fixed}, d\in\mathcal{D} & (32) \\
& \sum_{e\in\mathcal{E}} h_{edt} + \sum_{p\in\mathcal{P}} P_{pt}h_{dtp} = W_{td} & \forall t\in\mathcal{T}, d\in\mathcal{D} & (33) \\
& \sum_{e\in\mathcal{E}} x_{edt} + \sum_{p\in\mathcal{P}} P_{pt}n_p \ge N_{td} & \forall t\in\mathcal{T}, d\in\mathcal{D} & (34) \\
& p_{ed} \le \sum_{t\in\mathcal{T}} x_{edt} & \forall e\in\mathcal{E}, d\in\mathcal{D} & (35) \\
& h_{edt} \le 10x_{edt} \le max^{\text{day}} X_{edt} & \forall e\in\mathcal{E}, d\in\mathcal{D}, t\in\mathcal{T} & (36) \\
& x_{edt}, p_{ed} \in \{0,1\} & \forall e\in\mathcal{E}, d\in\mathcal{D}, t\in\mathcal{T} & \\
& n_{edt}, n_p, h_{dt} \in \mathbb{N}^+ & \forall e\in\mathcal{E}, d\in\mathcal{D}, t\in\mathcal{T}, p\in\mathcal{P} & \\
& \Pi_1^\alpha, \Pi_1^\beta, \Pi_1^\gamma, \Pi_1^\delta, \Pi_1^\varepsilon \in \mathbb{R}^+ & &
\end{array}
$$

MILP1

above. The determination of weights α_1 to ε_1 will be discussed in section 5.3.1. Constraint sets (29) to (31) are related to legal requirements. Constraint set (29) ensures that an employee cannot work less than min^{day} hours nor more than max^{day} hours during a working day. The total number of hours worked in a week cannot exceed max^{week} hours (constraint set (30)). Short-term contract employees cannot work more than 35 hours a week; on the other hand, hiring an employee for less than a week is not common and not easy. For these reasons, we make sure that all temporary workers work exactly 7 hours a day (constraint set (31)). As mentioned in section 5.1.3, some employees work under a special contract which makes their working time fixed instead of flexible, although their exact tasks during that time still have to be calculated. Constraint set (32) ensures that the total number of hours worked by those employees matches exactly their contract.

Constraint set (33) ensures that the total number of hours worked by regular employees and short term workers matches the workload need. Constraint (34) ensures that the number of persons needed at the same time is consistent for each task and day.

Constraint sets (35) and (36) define the links between x, p and h, followed by the non-negativity constraints. Constraint set (35) ensures that an employee is present on a given day if and only if he has tasks assigned for that day. The left-hand side of the inequality in constraint set (36) defines the number of hours worked by each employee per day, making sure that it cannot exceed the maximum number of daily hours required by the law. The right-hand side inequality of the constraint ensures that the abilities, availabilities and work planning constraints (as defined in the data matrix X) are met.

5.2.1.2 MILP2

Solving MILP1 gives the number of hours per employee and per day, and the number of temporary workers to be hired with their profiles. These data (h_{edt}, h_{dtp} and n_p) are then reprocessed in order to include the temporary workers in a new employee set $\mathcal{E}'$, such that $|\mathcal{E}'| = |\mathcal{E}| + \sum_{p \in \mathcal{P}} n_p$. The number of hours h_{edt} obtained at the end of

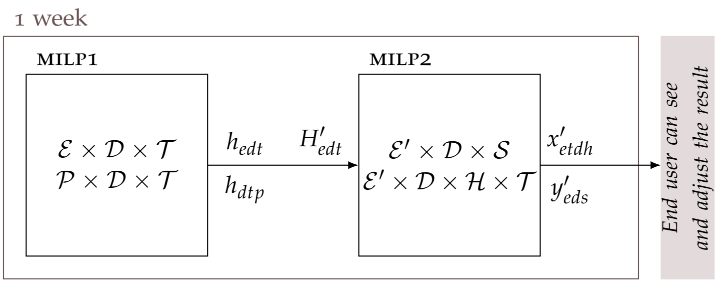


Figure 5.2: Weekly timetabling inputs and outputs

the first step is reprocessed to include the temporary workers as well, to become H'_{edt} (defined on $\mathcal{E}' \times \mathcal{D} \times \mathcal{T}$) and be used as an input for MILP2 – see Figure 5.2.

Sets $\mathcal{S}$ and $\mathcal{H}$, namely the shifts and hours, are now used besides the sets used in MILP1. MILP2 is thus refining the decisions made in MILP1 by considering the timetabling on a more detailed scale.

INPUT DATA. MILP2 uses the following input data:

– From the outputs of step 1:

H'_{edt} Number of hours worked by employee $e \in \mathcal{E}'$ on day $d \in \mathcal{D}$ and task $t \in \mathcal{T}$.

– Data describing the workload:

W^1_{tdh} Workload (in working hours) for task $t \in \mathcal{T}^1$ defined on a precise time window, for hour $h \in \mathcal{H}$ and day $d \in \mathcal{D}$.

W^2_{td} Workload (in working hours) of task $t \in \mathcal{T}^2$ defined on a slot, for day $d \in \mathcal{D}$.

S_{thd} Slot description: $S_{thd} = 1$ if task $t \in \mathcal{T}^2$ can be done on hour $h \in \mathcal{H}$ and day $d \in \mathcal{D}$, 0 otherwise.

Max_{tdh} Handling equipment upper bound. Max_{td} is the amount of handling equipment available for task $t \in \mathcal{T}$, day $d \in \mathcal{D}$ and hour $h \in \mathcal{H}$. Its value can be infinite.

– Description of the shifts:

Z_{sh} Shift description: $Z_{sh} = 1$ if hour $h \in \mathcal{H}$ is in shift $s \in \mathcal{S}$, 0 otherwise.

D_s Shift duration: D_s is the length of shift $s \in \mathcal{S}$, in hours.

DECISION VARIABLES. The aim of this step is to create a weekly schedule, giving for each employee (temporary workers included) their exact working times. Since the possible shifts are input data, the aim of the current step is to choose the right shift for each employee per day, ensuring that this allocation matches the workload needed. The decision variables used at this step are therefore the following:

x'_{etdh} Percentage of time spent on task $t \in \mathcal{T}$ by employee $e \in \mathcal{E}'$ on day $d \in \mathcal{D}$ and hour $h \in \mathcal{H}$.

y'_{eds} Shift allocation: $y'_{eds} = 1$ if employee $e \in \mathcal{E}'$ is allocated to shift $s \in \mathcal{S}$ on day $d \in \mathcal{D}$.

OBJECTIVE FUNCTION. The penalties which are part of the objective function are defined as follows.

Π^α_2 *Under/over-staffing* penalty. We give a penalty point each time a person is assigned in excess or missing for a task, compared to the needed workload.

Π^β_2 *Hour adjustments* penalty. The number of hours calculated in step 1 did not take the task slots into account; it could therefore need a few adjustments to have a feasible solution for MILP2.

We give a penalty point if we have to remove an hour from the working time calculated for an employee on a given day.

Π_2^γ *Handling equipment* penalty. Knowing the upper bound on the amount of handling equipment, we give a penalty point each time we have to rent an extra machine during one hour to be able to perform a task.

MODEL. MILP2 is written as follows.

$$
\begin{array}{lllr}
\min & \alpha_2\Pi_2^\alpha + \beta_2\Pi_2^\beta + \gamma_2\Pi_2^\gamma & & \\
\text{s.t.} & \Pi_2^\alpha = \sum_{t\in\mathcal{T}^1,d\in\mathcal{D},h\in\mathcal{H}} \left|W_{tdh}^1 - \sum_{s\in\mathcal{S},e\in\mathcal{E}'} Z_{sh}y'_{eds}\right| & & \\
& \qquad\qquad + \sum_{t\in\mathcal{T}^2,d\in\mathcal{D}} \left|W_{td}^2 - \sum_{h\in\mathcal{H},e\in\mathcal{E}} S_{thd}x'_{etdh}\right| & & (37) \\
& \Pi_2^\beta = \sum_{e\in\mathcal{E},d\in\mathcal{D},t\in\mathcal{T}} \left(H'_{edt} - \sum_{h\in\mathcal{H}} x'_{etdh}\right) & & (38) \\
& \Pi_2^\gamma = \sum_{t\in\mathcal{T},d\in\mathcal{D},h\in\mathcal{H}} \left(\sum_{e\in\mathcal{E}} x'_{etdh} - Max_{tdh}\right) & & (39) \\
& \sum_{s\in\mathcal{S}} D_s y'_{eds} = \sum_{t\in\mathcal{T}} H'_{edt} & \forall e\in\mathcal{E}', d\in\mathcal{D} & (40) \\
& \sum_{t\in\mathcal{T}} x'_{etdh} = \sum_{s\in\mathcal{S}} Z_{sh}y'_{eds} & \forall e\in\mathcal{E}', d\in\mathcal{D}, h\in\mathcal{H} & (41) \\
& \sum_{s\in\mathcal{S}} y'_{eds} \leq 1 & \forall e\in\mathcal{E}', d\in\mathcal{D} & (42) \\
& x'_{etdh} \leq 1, x'_{etdh} \in \mathbb{R}^+ & \forall e\in\mathcal{E}', t\in\mathcal{T}, d\in\mathcal{D}, h\in\mathcal{H} & \\
& y'_{eds} \in \{0,1\} & \forall e\in\mathcal{E}', d\in\mathcal{D}, s\in\mathcal{S} & \\
& \Pi_2^\alpha, \Pi_2^\beta, \Pi_2^\gamma \in \mathbb{R}^+ & &
\end{array}
$$

MILP2

Constraint sets (37) to (39) define the penalties as described above. We note that constraint set (37) matches the employees' presence with the workload need, both for the tasks $t \in \mathcal{T}^1$ defined per hour and the tasks $t \in \mathcal{T}^2$ defined per slot.

Besides constraint set (38), constraint set (40) also ensures the continuity with step 1, by matching the length of the shifts with the number of hours per employee defined by MILP1. Constraint set (41) defines the link between x', y' and Z, while set (42) makes sure that each employee has no more than one task per hour.

5.2.2 *Daily rostering:* MILP3

The aim of this step is to build the detailed schedule for a given day of the week. While MILP1 and MILP2 are meant to be used every week to set up the following week's planning, this third model is supposed to be run every morning to plan the upcoming day.

Depending on the rostering requirements, the time scale can be further refined to use time windows smaller than an hour (in our tests for example, we use 15 minutes time windows). Recall from section 5.1.3 that these time windows are called intervals and defined on the set $\mathcal{Q}$. The outcome of MILP3 is thus an assignment of tasks to employees for each interval.

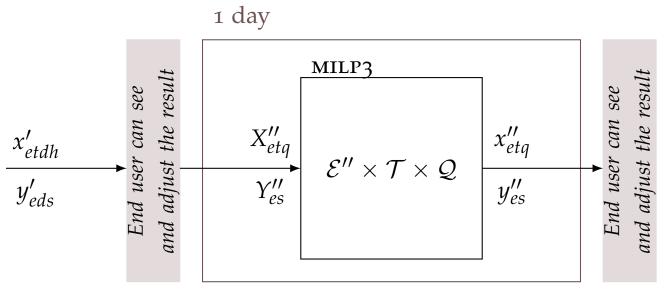


Figure 5.3: Daily rostering inputs and outputs

Some information may be adjusted before running MILP3. For instance, the temporary workers suggested by MILP1 have been hired and we now know their exact qualifications; the employees who are absent do not need to be considered; the manager also has the possibility to make a few changes in the timetable produced by MILP2. The new set of employees is therefore noted $\mathcal{E}''$. The data about the workload also evolves; it is more precise than (and possibly very different from) previous available forecasts.

The interval structure and the given day being known, the output of MILP2 x'_{etdh} is reprocessed to obtain the matrix X'' defined on $\mathcal{E}'' \times \mathcal{T} \times \mathcal{Q}$. X''_{etq} gives the percentage of time spent on task t by employee e during interval q, as calculated by MILP2 and possibly adjusted by the manager if needed. Similarly, the output of MILP2 y'_{eds} are reprocessed into the matrix Y''_{es} defined on $\mathcal{E}'' \times \mathcal{S}$, which contains the shift allocations planned for the given day. The other input data needed are similar to the ones used in the previous steps, but they are now defined on intervals rather than hours, and for the given day only. Figure 5.3 summarizes the mechanism of MILP3.

INPUT DATA. To summarize, the following data is used as inputs for MILP3:

- X_{etq} Binary matrix similar to X_{edt} described in section 5.2.1.1. $X_{etq} = 1$ if employee $e \in \mathcal{E}''$ is present on time interval $q \in \mathcal{Q}$, able to do task $t \in \mathcal{T}$, and if task t can be done on interval q; 0 otherwise.
- X''_{etq} Task allocation as calculated by MILP2 and possibly adjusted by the manager: $X''_{etq} = 1$ if employee $e \in \mathcal{E}''$ is allocated to task $t \in \mathcal{T}$ during time interval $q \in \mathcal{Q}$; 0 otherwise.
- Y''_{es} Shift allocation as calculated by MILP2 and possibly adjusted by the manager: $Y''_{es} = 1$ if employee $e \in \mathcal{E}''$ is allocated to shift $s \in \mathcal{S}$, 0 otherwise.
- W^1_{tq} Workload for task $t \in \mathcal{T}^1$ defined per interval, for time interval $q \in \mathcal{Q}$.
- W^2_t Workload of the considered day for the task $t \in \mathcal{T}^2$ defined per slot.

S_{tq} Slot description: $S_{tq} = 1$ if task $t \in \mathcal{T}^2$ can be done during interval $q \in \mathcal{Q}$, 0 otherwise.

Max_{tq} Handling equipment upper bound: Max_{tq} is the number of machines available for task $t \in \mathcal{T}$ during interval $q \in \mathcal{Q}$. This value can be infinite.

Max_t Maximum amount of time (in intervals) that a worker can spend per day on task $t \in \mathcal{T}$. This value enables one to respect safety and ergonomics principles.

DECISION VARIABLES. This step uses two decision variables:

y''_{es} Shift allocation: $y''_{es} = 1$ if employee $e \in \mathcal{E}''$ is allocated on shift $s \in \mathcal{S}$, 0 otherwise.

x''_{etq} Task allocation: $x''_{etq} = 1$ if employee $e \in \mathcal{E}''$ works on task $t \in \mathcal{T}$ during interval $q \in \mathcal{Q}$, 0 otherwise.

OBJECTIVE FUNCTION. The following penalties are part of the objective function:

Π_3^α *Shift changes* penalty. A penalty point is given for each employee whose shift has been changed, compared to the plan made at the end of MILP2.

Π_3^β *Task changes* penalty. A penalty point is given each time the task of an employee is changed compared to what was planned at the end of MILP2.

Π_3^γ Knowing the *handling equipment* upper bound, a penalty is given for each interval for which an additional handling machine has to be rent to be able to perform a task.

Π_3^δ *Ergonomy* penalty. A penalty point is given for each interval in excess for a worker, compared to the maximum amount of time per day defined for his task.

MODEL. MILP3 is expressed as shown on the following page. Constraint sets (43) to (46) define the penalties (shift changes, task changes, handling equipment, ergonomy) as described above. Constraint sets (47) and (48) match the workers to the workload, for the tasks from $\mathcal{T}^1$ defined per hour and for the tasks from $\mathcal{T}^2$ defined by slots, respectively. Constraint set (49) checks that the tasks are allocated to the employees only when it is possible. Finally, constraint sets (51) and (52) ensure that each employee has no more than one shift per day, and one task per interval.

$$
\begin{array}{lll}
\min & \alpha_3\Pi_3^{\alpha} + \beta_3\Pi_3^{\beta} + \gamma_3\Pi_3^{\gamma} + \delta_3\Pi_3^{\delta} & \\
\text{s.t.} & \Pi_3^{\alpha} = \sum_{e\in\mathcal{E}'',s\in\mathcal{S}} \left|Y''_{es} - y''_{es}\right| & (43)\\
& \Pi_3^{\beta} = \sum_{e\in\mathcal{E}'',t\in\mathcal{T},q\in\mathcal{Q}} \left|X''_{etq} - x''_{etq}\right| & (44)\\
& \Pi_3^{\gamma} = \sum_{t\in\mathcal{T},q\in\mathcal{Q}} \left(\sum_{e\in\mathcal{E}''} x''_{etq}\right) - \text{Max}_{tq} & (45)\\
& \Pi_3^{\delta} = \sum_{e\in\mathcal{E}'',t\in\mathcal{T}} \left(\sum_{q\in\mathcal{Q}} x''_{etq}\right) - \text{Max}_{t} & (46)\\
& \sum_{e\in\mathcal{E}''} x''_{et_1q} = W^1_{t_1q} \qquad \forall t\in\mathcal{T}^1, q\in\mathcal{Q} & (47)\\
& \sum_{e\in\mathcal{E}'',q\in\mathcal{Q}} S_{t_2q}x''_{et_2q} = W^2_{t_2} \qquad \forall t\in\mathcal{T}^2 & (48)\\
& x''_{etq} \leq X_{etq} \qquad \forall e\in\mathcal{E}'', t\in\mathcal{T}, q\in\mathcal{Q} & (49)\\
& x''_{etq} = \sum_{s\in\mathcal{S}} Z_{sq}y''_{es} \qquad \forall e\in\mathcal{E}'', t\in\mathcal{T}, q\in\mathcal{Q} & (50)\\
& \sum_{t\in\mathcal{T}} x''_{etq} \leq 1 \qquad \forall e\in\mathcal{E}'', q\in\mathcal{Q} & (51)\\
& \sum_{s\in\mathcal{S}} y''_{es} \leq 1 \qquad \forall e\in\mathcal{E}'' & (52)\\
& x''_{etq}, y''_{es} \in \{0,1\} \qquad \forall e\in\mathcal{E}'', t\in\mathcal{T}, q\in\mathcal{Q} & \\
& \Pi_3^{\alpha}, \Pi_3^{\beta}, \Pi_3^{\gamma}, \Pi_3^{\delta} \in \mathbb{R}^+ &
\end{array}
$$

MILP3

5.2.3 *Complexity*

The values of all elements are between $\frac{B}{4}$ and $\frac{B}{2}$.

In this section, each step of the problem is shown to be NP-hard in the strong sense, by a transformation from the 3-partition problem already described in section 2.2.2. Recall from section 2.2.2 that the 3-partition problem consists in dividing $3n$ elements r_i whose sum is Bn into n groups of sum B. If such a partition exists, each group (each subset A_j with $j \in \{1, 2, ..., n\}$) contains exactly 3 elements. Garey and Johnson [77] have shown that the problem is NP-hard in the strong sense.

INSTANCE OF THE TIMETABLING PROBLEM (STEP 1).

Let us consider an instance of step 1 with:

- one day: $\mathcal{D} = \{1\}$;
- $3n$ tasks: $\mathcal{T} = \{1, ..., 3n\}$;
- n employees, none of them having a fixed shift: $\mathcal{E} = \{1, ..., n\}$, $\mathcal{E}^{\textit{fixed}} = \emptyset$; all employees are available on day 1 ($X_{edt} = 1$ for all $e \in \mathcal{E}, d \in \mathcal{D}, t \in \mathcal{T}$) and perfectly skilled for all tasks ($Q_{et} = \zeta$ for all $e \in \mathcal{E}, t \in \mathcal{T}$);
- one temporary worker profile ($\mathcal{P} = \{1\}$): the profile is one of a worker skilled on all tasks ($P_{1t} = 1$ for all $t \in \mathcal{T}$) and hiring such a worker costs 1 unit ($C_1 = 1$);
- a working day that cannot be longer than B hours: $min^{\text{day}} = 0$, $max^{\text{day}} = B$, $max^{\text{week}} = B$;
- there are no safety or ergonomics constraints: $Max_t = \infty$;
- a workload that matches the integers given as data in the 3-partition problem: $W_{td} = r_t$ for day $d \in \mathcal{D}$ and all tasks $t \in \mathcal{T}$.

Proposition. There exists a 3-partition if and only if there exists a solution to the corresponding instance of step 1 of our timetabling problem with a cost 0.

Proof. Necessity. Suppose there exists a 3-partition $\{A_1, A_2, ..., A_n\}$. Let us build a solution to step 1 with a total cost of zero. Since all employees are qualified for all tasks, it is possible to allocate the three tasks indexed by the elements of set A_j to employee j. By definition of the 3-partition, $\sum_{i \in A_j} r_i = B$, thus B hours are needed to complete these three tasks: employee j works B hours on day 1, which respects the legal constraints. This way, all tasks can be allocated with no need for temporary workers, thus $\Pi_1^{\alpha} = 0$. All employees are perfectly qualified and planned to be present, thus the qualification penalty and the unplanned absence penalty are equal to zero. So is the equity penalty since all workers work exactly the same amount of time (B hours). The ergonomic penalty is also zero since there are no ergonomic constraints. The total cost of the solution is therefore zero.

Sufficiency. Suppose that a solution of cost zero exists for step 1; let us show that a 3-partition exists. A cost equal to zero means that no temporary worker is hired in this solution: therefore all the workload (equal to $\sum_{t \in \mathcal{D}, d \in \mathcal{D}} W_{td} = \sum_{t \in \{1,...,3n\}} r_t = Bn$) has been divided among regular employees. Because the equity penalty and the unplanned absence penalty are both equal to zero, the workload Bn is equally distributed between the n employees. Each employee therefore works B hours in the solution considered. This provides, for all employees $j \in \{1, 2, ..., n\}$, a partition of tasks into triples $\{A_1, A_2, ..., A_n\}$ such that $\sum_{i \in A_j} r_i = B$. $\square$

The same proof can be done for step 2 by refining the same instance as follows. From step 1 it comes that $H'_{edt} = B$ for all employees, days and tasks. The planning horizon counts B hours, thus $\mathcal{H} = \{1, ..., B\}$. The data can be refined by setting $\mathcal{T}^1 = \mathcal{T}$ and $\mathcal{T}^2 = \emptyset$. Only one shift is needed that lasts B hours: $\mathcal{S} = 1$, $Z_{1h} = 1$ for all hours $h \in \mathcal{H}$ and $D_1 = 1$. Finally $Max_{ti} = \infty$.

The same demonstration is also valid for step 3 by setting one-hour long intervals ($\mathcal{H} = \mathcal{Q}$) and $Max_{ti} = Max_t = \infty$.

5.3 NUMERICAL RESULTS

In this section, the linear programs detailed previously are tested to assess their performances in different situations. The results described in section 5.3.1 have been obtained using industrial data, while section 5.3.2 details the results obtained on a benchmark data set made available by Günther and Nissen [94, 95]. The results in section 5.3.3 are based on instances generated for testing purposes.

5.3.1 *Testing industrial instances*

The models were tested on data provided by our industrial partner, using CPLEX for academic purposes and a free integer programming solver for industrial use.

5.3.1.1 *Instance description*

30 different data sets from real-life industrial cases were tested. Three different warehouse teams with different configurations were considered, and the decision-support tool was tested for 10 weeks in each team. The different configurations are as follows:

CONFIGURATION 1 has 5 days, 16 hours, 64 intervals, 17 possible shifts, 19 tasks, 11 employees, 18 temporary workers profiles. This warehouse team has only one big client. The products arrive in containers from overseas, and the arrival of containers creates a peak of activity for the unloading-related tasks. One single task can therefore represent the major part of the workload, which makes the workload distribution geometric. The execution time is 5 seconds on the average for this configuration.

CONFIGURATION 2 has 5 days, 16 hours, 64 intervals, 12 possible shifts, 44 tasks, 15 employees, 1 temporary worker profile. This warehouse has several smaller clients. It makes the number of tasks higher, and statistically smooths the occurrences of unloading and preparation tasks, therefore the workload distribution is normal. It does not vary much from one day to another, therefore this team almost never uses temporary workers – this is why they define only one temporary worker profile. The execution time is 10 seconds on the average for this configuration.

CONFIGURATION 3 has 6 days, 12 hours, 24 intervals, 36 possible shifts, 13 tasks, 2 employees, 8 temporary worker profiles. This activity is only seasonal, therefore the 2 regular employees handle the management tasks, and the operational work is done by up to 200 temporary workers of 8 different profiles. The execution time is 8 seconds on the average for this configuration.

Two instances, typical of configurations 1 and 2, and made anonymous for confidentiality concerns, are available at `www.g-scop.fr/~gaujalg/TimeTabling`.

5.3.1.2 *Numerical results with* CPLEX

The results obtained on the two instances available online are displayed in Table 5.1 – the figures for MILP3 are the mean of the results obtained for the five days. More details on solutions, including the penalty points and results on each day of the week (via MILP3), can

	Execution time (seconds)			Value of the obj. function		
Instance	MILP1	MILP2	MILP3	MILP1	MILP2	MILP3
`instance1`	2.57	1.53	0.05	5 900	40 388	704
`instance2`	5.62	2.05	0.14	8 290	35 274	889

Table 5.1: Results obtained on `instance1` and `instance2`

be found online along with the instances, together with a graphical interface that enables one to visualize the instances. Another version of this interface has been developed, that permits to run the three sequential models using CPLEX. It is used for teaching purposes, to let students build their own timetabling models by a trial and error approach.

See some screenshots of this interface in Appendix E.

5.3.1.3 *Results in the industrial context*

Our industrial partner favored a free integer programming solver, for economical reasons. Thus for company use, our methods are implemented with lp_solve 5.5.2, a free MILP solver under the GNU LGP license. With CPLEX, for all the instances tested, the computing times were below 10 seconds. The computation times are much higher with lp_solve for the same instances. For practical reasons, the computation of lp_solve is interrupted after a short period of time, and we keep the best feasible solution found during its search. For our instance1, setting up a time-out at 20 seconds gives a 40% gap to optimal, and waiting for 2 minutes reduces this gap to 15% for MILP1 (which is the longer step in terms of execution time) – see Figure 5.4. The company was satisfied with the quality of the solution obtained at this point, even though it is not optimal.

The GNU Lesser General Public License allows companies to integrate software into their own proprietary software – see `www.gnu.org/copyleft/lesser`.

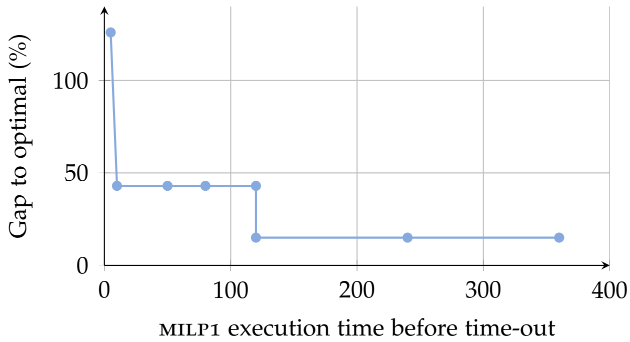


Figure 5.4: Gap to optimal with time-out on lp_solve

In our industrial context, the logistics provider was satisfied with the quality of the timetable the managers could obtain manually; the problem was that the operation was extremely long and tedious. The main objective was therefore to automate that process, but keeping an outcome similar to the one the manager would have obtained manually. The soft constraints were created in this perspective: listing the criteria that make a timetable better than another from the manager's point of view. These soft constraints being weighted in our objective functions, the manager has the possibility to play with the parameters until he gets the solution he is most satisfied with. The managers who are experienced in manual timetabling can therefore choose the settings with a trial and error approach.

From a theoretical point of view, using a weighted objective function raises some issues about the best way to fix the parameters for non-homogeneous criteria – see Pöyhönen and Hämäläinen [162]. Methods like UTA (additive utilities) or AHP (Analytical Hierarchy Process) can be used to determine the weights; the interested reader can refer to the survey on multiple criteria decision analysis by Figueira *et al.* [72].

Here, the weights in the objective function are determined in an iterative process, using interviews and expert opinions, together with the trial and error test runs, prior to real planning runs. During these test runs, the weights are set such that the solution proposed by our tool is not very different from what a manager would have planned himself. We note that such iterative procedures can be found in the literature for industrial applications (see Günther and Nissen [94]). In this practical situation, the fact that the weight settings are not fixed is perceived as an asset by the end user, who uses them as an actual managing tool. It offers more flexibility, for instance to change the relative importance of the soft constraints depending on the nature of the upcoming activity.

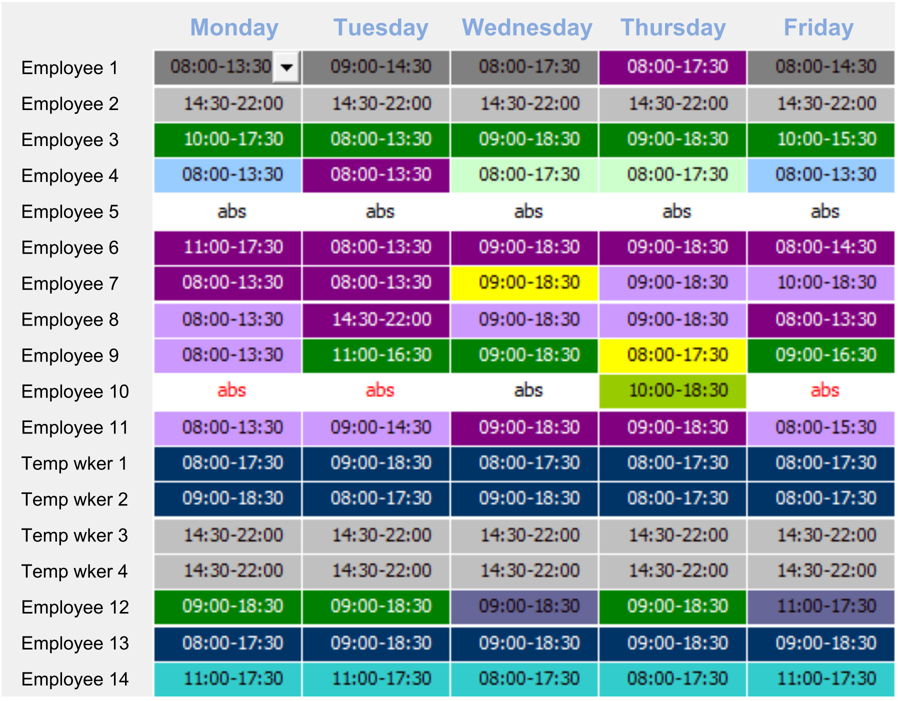
The outputs are the weekly timetable (see Figure 5.5 as an example) and the daily roster (Figure 5.6). They are presented in a table format, ready to be used by the manager. The value added compared to the former situation is the speed of the timetable operation: the automated timetable is generated in a few seconds, whereas the manual process was tedious and time consuming and took up to four hours.

5.3.2 *Benchmark*

In order to assess the performance of our model compared to existing benchmarks, we tested it on the instances provided by Günther and Nissen [94, 95]. This benchmark [1] comes from a real-world logistics case. Since the context is similar to ours, this data set matches the main characteristics of our problem, as described in section 5.1.2.1: very diverse skills, flexible working hours, uneven workload.

With 65 employees, 9 tasks and 13 shift types to be scheduled over 7 days, this set of data is different from the instances tested in sec-

1. available at `http://www.tu-ilmenau.de/fileadmin/media/wid/forschung/TestproblemePersonaleinsatzplanung/`



	Monday	Tuesday	Wednesday	Thursday	Friday
Employee 1	08:00-13:30	09:00-14:30	08:00-17:30	08:00-17:30	08:00-14:30
Employee 2	14:30-22:00	14:30-22:00	14:30-22:00	14:30-22:00	14:30-22:00
Employee 3	10:00-17:30	08:00-13:30	09:00-18:30	09:00-18:30	10:00-15:30
Employee 4	08:00-13:30	08:00-13:30	08:00-17:30	08:00-17:30	08:00-13:30
Employee 5	abs	abs	abs	abs	abs
Employee 6	11:00-17:30	08:00-13:30	09:00-18:30	09:00-18:30	08:00-14:30
Employee 7	08:00-13:30	08:00-13:30	09:00-18:30	09:00-18:30	10:00-18:30
Employee 8	08:00-13:30	14:30-22:00	09:00-18:30	09:00-18:30	08:00-13:30
Employee 9	08:00-13:30	11:00-16:30	09:00-18:30	08:00-17:30	09:00-16:30
Employee 10	abs	abs	abs	10:00-18:30	abs
Employee 11	08:00-13:30	09:00-14:30	09:00-18:30	09:00-18:30	08:00-15:30
Temp wker 1	08:00-17:30	09:00-18:30	08:00-17:30	08:00-17:30	08:00-17:30
Temp wker 2	09:00-18:30	08:00-17:30	09:00-18:30	08:00-17:30	08:00-17:30
Temp wker 3	14:30-22:00	14:30-22:00	14:30-22:00	14:30-22:00	14:30-22:00
Temp wker 4	14:30-22:00	14:30-22:00	14:30-22:00	14:30-22:00	14:30-22:00
Employee 12	09:00-18:30	09:00-18:30	09:00-18:30	09:00-18:30	11:00-17:30
Employee 13	08:00-17:30	09:00-18:30	09:00-18:30	09:00-18:30	09:00-18:30
Employee 14	11:00-17:30	11:00-17:30	08:00-17:30	08:00-17:30	11:00-17:30

Figure 5.5: Weekly timetable used by the manager

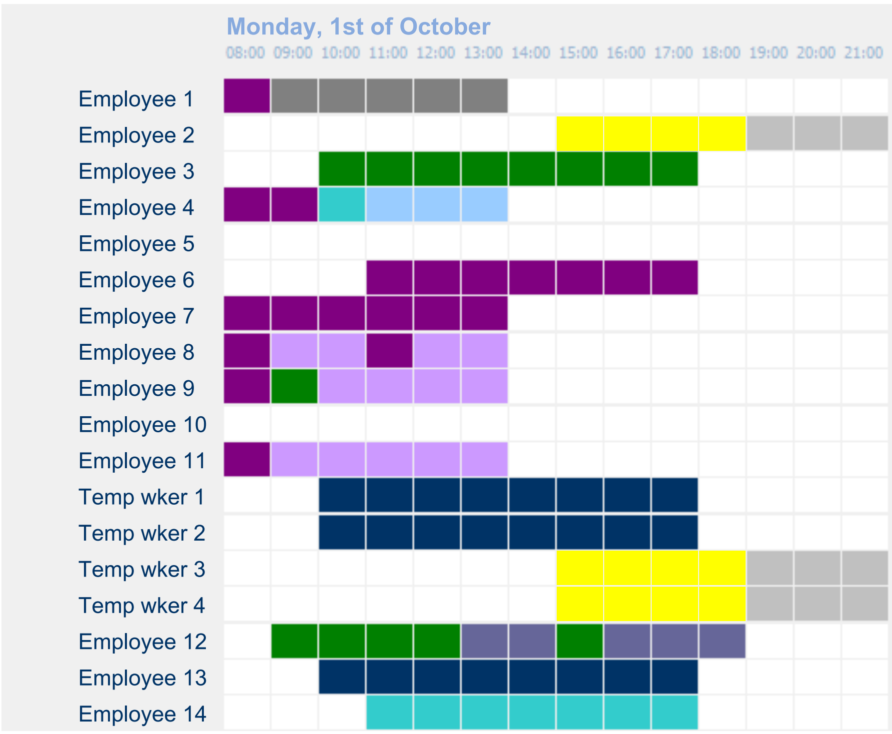


Figure 5.6: Daily roster used by the manager
Each color represents a task.

tion 5.3.1. It has half as many tasks but up to 6 times more employees, and the scheduling period is two days longer. Note that the objective function used by Günther and Nissen is also different. They penalize under-staffing and the number of job changes as we do, but they distinguish between two cases for over-staffing: over-staffing at a period when the workload needed (*i. e.* demand) is zero is seen as much worse than over-staffing when there is a positive demand. All the other criteria we used in our objective functions are not taken into account by Günther and Nissen. This is due to how their model constraints are related to the daily rostering only, whereas we use three different models with different time scales in our approach. Solving three MILPs sequentially requires extra constraints to link the different steps with each other (*e. g.* the output of MILP1 is included in a constraint in MILP2).

For this reason, we decided to run our model keeping the same coefficients of the objective function used in the previous section. Hence, it is important to note that we do not optimize exactly the same criteria as Günther and Nissen. This test only aims at checking that we can process a different type of data in a reasonable amount of time. Nevertheless, in order to have some comparison points, we reprocessed our output in order to evaluate it with Günther and Nissen's criteria. The results are shown in Table 5.2.

	Execution time	over-staffing when no demand	over-staffing when demand > 0	under-staffing	number of job changes
Our results	2.59 min	7 995	795	0	664
Günther's manual timetabling	unknown	33 795	14 610	20 130	0
Günther's best solution	about 50 min	7 245	28 395	7 355	1 502

Table 5.2: Comparison with Günther and Nissen's results

The differences observed in Table 5.2 in terms of objective function values is due to the differences in the constraints of the two models. Our model does not allow under-staffing at the daily level, which explains why we get 0 on that criterion. Günther and Nissen's model also gives work to each and every employee, while our model can assign days off rather than creating over-staffing. But we see that our sequential method gives results in about 3 minutes, while Günther and Nissen's particle swarm optimization requires an execution time of about 50 minutes. It meets our goal of providing logistic managers with a decision tool that can be used daily without a long execution time.

5.3.3 *Numerical results from the generated instances*

The previous section demonstrates that the weekly and daily timetables for real size problems can be solved easily and in a fast manner.

Real-life problems are actually very constrained: if some employees have shifts fixed beforehand, or are absent for one or two days, the solution space is reduced and the problem is easier to solve. The goal of the current section is to show the performance and limits of the models detailed previously, regarding some of the input parameters that can make the situation more complicated.

The focus is especially set on the workload variations and the workforce abilities. Since workload variations are one of the main challenges warehouses have to cope with, it is important to check how the model behaves when the workload changes. One of the main tools to absorb these variations is the adjustment of the workforce skills; therefore, this piece of data is also varied in our test. One goal is to see what could be a good policy for the employees' training in the warehouse.

The weights linked to soft constraints which are not related to the workload (β_1, γ_1, δ_1 ε_1, γ_2, γ_3 and δ_3) are put to zero. The remaining

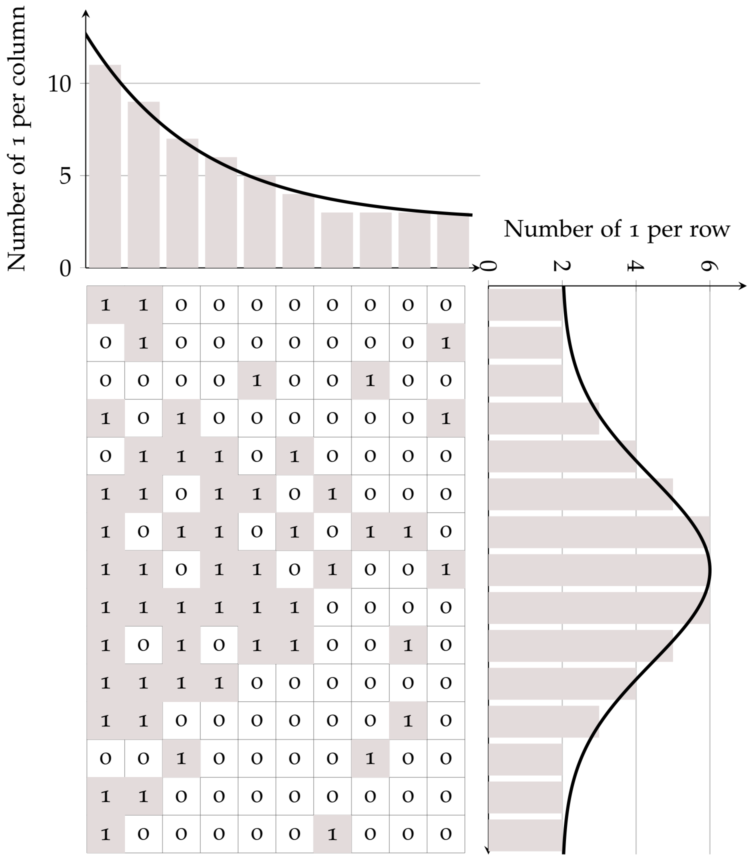


Figure 5.7: Example of matrix P_{pt}
Geometric distribution on one dimension, normal distribution on the other

weight parameters are assumed to be equally important, and hence fixed such that $\alpha_1 = \alpha_2 = \beta_2 = \alpha_3 = \beta_3 = 1$.

The data matrices representing the workload (W^2_{td}), the temporary workers profiles (P_{pt}) and the abilities (part of X_{edt} or X_{eti}) are generated randomly. Each dimension of the given matrix is generated using a normal, geometric or uniform distribution function. As an example, Figure 5.7 on the previous page displays a temporary workers profile matrix P_{pt}. In this figure, the number of tasks mastered by each employee (given in the rows) follows a normal distribution, while the number of employees who can perform each task (given in the columns) is randomly drawn from a geometric distribution. Similarly, different P_{pt} are generated by changing the type of distribution (normal, geometric or uniform) for the columns or the rows.

Considering that a temporary worker is likely to be paid more if he has more skills, the temporary workers cost C_p are defined as the number of tasks mastered by the worker p.

The workload is assumed to be entirely defined per slots, *i. e.* $\mathcal{T}^1 = \varnothing$ and $\mathcal{T}^2 = \mathcal{T}$. The slot description matrix S is set such as $S_{thd} = 1$ for all $t \in \mathcal{T}$, $h \in \mathcal{H}$, $d \in \mathcal{D}$. Once W^2_{tdh} is generated as described above, the workload W_{td} and the number N_{td} of people needed at the same time are deduced easily.

As seen with the industrial case study, most of the input parameters (*e. g.* fixed work times F_{ed} and planned absences) are context dependent. Therefore, such input parameters (listed below) are fixed in all tested instances:

X_{edt} Data matrix containing the abilities, availabilities and work planning information. The planned absences narrow the decision space; therefore all the employees $e \in \mathcal{E}$ are assumed to be available for all time units $t \in \mathcal{T}$ and $q \in \mathcal{Q}$. Similarly, all tasks $t \in \mathcal{T}$ need to be carried out during all the time intervals considered.

Q_{et} Non-binary qualifications matrix. This matrix only plays a role in one soft constraint (penalty Π^{α}_1 defined in constraint (24)). For the sake of simplicity, only binary qualifications are used for our tests, thus $Q_{et} = 1$ if the employee e is qualified for the task t, 0 otherwise.

Max_t Safety and ergonomics upper bound. This vector is also only related to soft constraints. We relax it setting $Max_t = \infty$ for all $t \in \mathcal{T}$.

Max_{tdh}, Max_{tq} Handling equipment upper bounds. Similarly, these matrices are only related to soft constraints. We relax them setting $Max_{tdh} = Max_{tq} = \infty$ for all $t \in \mathcal{T}$, $d \in \mathcal{D}$, $h \in \mathcal{H}$, $q \in \mathcal{Q}$.

F_{ed} Fixed work times. Defining some shifts beforehand also narrows the decision space, thus we set $\mathcal{E}^{fixed} = \varnothing$, which means that F_{ed} does not need to be defined.

Z_{sh} Shift description. This matrix is built enumerating all legally possible shifts. The shift duration D_s is easily obtained from the matrix Z.

5.3.3.1 *Tests on weekly timetabling*

One set of tests is carried out on MILP1 to study the impact of the abilities matrix shape on the execution time. The size of the instances is as follows: 10 employees, 8 tasks, 8 temporary worker profiles, 5 days, 24 hours, 115 possible shifts. 30 different abilities matrices are generated, using normal, geometric and uniform distributions. The normal distribution models a case where 80% of the employees can handle 20% of the tasks; it is the situation encountered most often in the real-life cases studied. The uniform distribution represents a case which can be seen as fairer: all the employees master the same number of tasks. The geometric distribution models a situation where most of the workers are hired for short periods of time, and therefore master only a few tasks. The organization relies upon a very small number of long-term employees to handle all the complex tasks.

Each matrix is tested with 40 different workload distributions. Each dot on Figure 5.8 shows the average computation time of MILP1 for these 40 different runs for a given abilities matrix. This average computation time is displayed as a function of the total number of abilities in the ability matrix.

The result is shown in Figure 5.8. On one extreme, when the ability matrix is near empty (low total number of abilities), the problem size is highly reduced, therefore the solution is obtained faster. On the other extreme, when the ability matrix is nearly full (high total number of abilities), it is easy to find a solution without any temporary workers, so the problem can be solved quickly as well. The most complex problems are for a half-full ability matrix, because the problem in this situation can become highly combinatorial. The instances generated from a geometric distribution regarding the tasks are especially difficult, because they contain one or two tasks, which

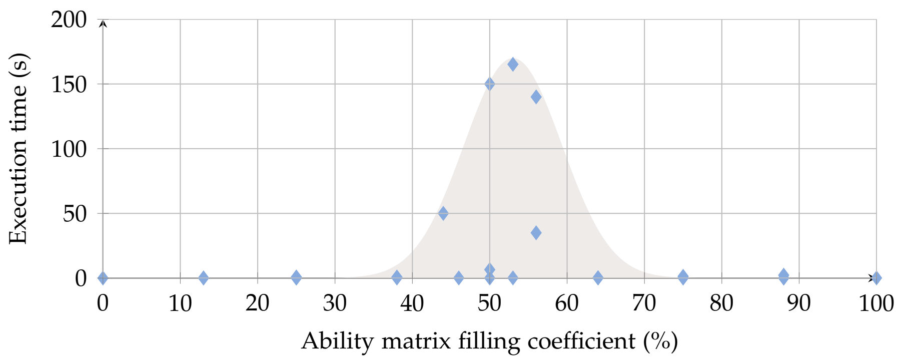


Figure 5.8: Execution time with different ability matrices

have a high workload compared to the others: if many employees are qualified for these tasks, there are many possible combinations for the allocation of tasks to employees. Therefore the search of a solution can fail; some of these instances cannot be solved with our test configuration due to a memory overflow.

On the contrary, the instances generated from normal distributions are faster to solve. Note that the computation times needed to solve the industrial instances were short, because the real-life data distributions were close to normal ones.

Since the other penalties are left aside, the only element optimized in the objective function of MILP1 is the number of temporary workers hired. Figure 5.9 shows how the total number of abilities influences this objective. It shows that training multi-skilled employees brings a real added value for the company up to 50% of the qualification matrix. It offers no further benefits, however, to increase this rate from 60% to 70% or higher.

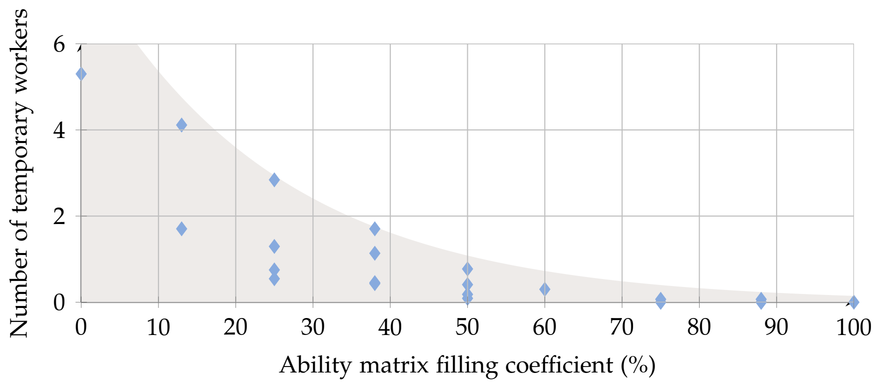


Figure 5.9: Temporary workers with different ability matrices

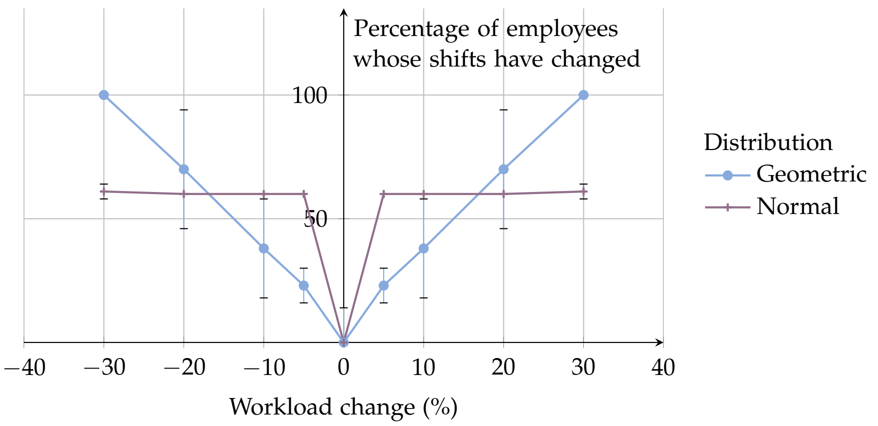
MILP1 determines the number of hours worked by each employee per day and per task. MILP2 refines this decision by choosing the exact shifts (starting and ending times) for each employee. Since the soft constraints are left aside and the workload used is only defined by slots, the problem modeled by MILP2 can be solved with a short computation time. With the test settings described previously, MILP2 can be solved in less than one second on all generated instances.

5.3.3.2 *Tests on daily rostering*

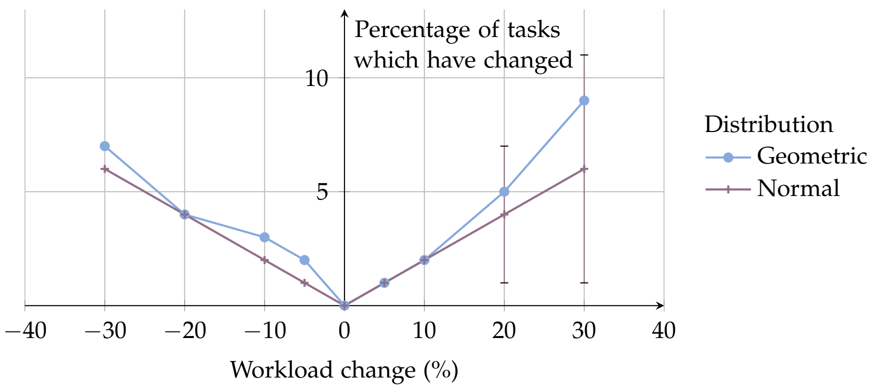
As explained in section 5.2.2, MILP3 is meant to be run every morning to plan the upcoming day, with input data that can possibly be very different from the ones used in the weekly timetable generation. Therefore the most important criterion to assess for MILP3 is the sensibility of the outcome, when the input data change between runs. For this set of tests, MILP1 and MILP2 are therefore run in sequence, with a set of instances which can be computed in a reasonable time (ability matrix filled to 60%), and different workload distributions (geometric and normal). Those were the two distribution types observed in our real-life situations, as explained in section 5.3.1 (see configurations 1 and 2). The workload matrix W^2_{ti} is created by increasing or reducing the workload used in MILP2 (W^2_{tdh}) for each task by a given percentage. MILP3 is solved with this new workload as an input and with all the other input data left unchanged. We then look at the values of Π^α_3 (number of employees whose shifts have been changed, compared to the plan made at the end of MILP2) and Π^β_3 (total number of times an employee's task has been changed, compared to what was planned at the end of MILP2). The length of an interval is set to 15 minutes, which means that the planning horizon considered has 96 intervals in total.

The results of these tests are shown in Figure 5.10 for Π^α_3 and Π^β_3, respectively. For improved readability, the values have been turned into percentages. Each dot is an average obtained from the results of 20 different instances, and the vertical lines show the corresponding standard deviations.

Figure 5.10a shows a linear increase of the number of changed shifts for the geometric workload distributions: with 30% change in



(a) Shift changes



(b) Tasks changes

Figure 5.10: MILP3 sensitivity when the workload changes

the workload, all the shifts need to be changed. Normal distributions are very sensitive to small workload changes, and reach stability very fast. Changing 50% of the shifts can handle a workload change of 5% to 30%. As shown in Figure 5.10b, the tasks changes stay below 10%: they are more constrained, since the qualification matrix does not allow to allocate an employee to any task. The number of task changes increases linearly for both distributions, which behave similarly for up to 10% of changes. For higher percentages, the normal distributions are the most robust again.

The daily schedule recreated by MILP3 from the weekly timetable can therefore stay close to the original plan, if the changes made in the workload are below 30%. Geometric distributions are more sensitive, since they contain one or two tasks which are very heavy compared to the others, and thus more difficult to redistribute without changing the employees' pre-planned shifts.

In the industrial context, it would thus be interesting to have workload distributions that are rather normal than geometric. The manager could try to negotiate with his client in order to get a better forecast of the upcoming volumes: since a better forecast leads to a better timetable, it can also help reducing the costs the client is charged for. The other option for the manager is to try and consolidate the flows, for instance by having several clients served by the same warehousing team, in order to further smoothen the workload.

5.4 CONCLUSION

This chapter proposes a decomposition approach to solve complex timetabling problems on different time scales (weekly and daily). Three Mixed and Integer Linear Programs are solved sequentially in order to achieve both weekly timetabling and daily rostering in an integrated manner. The proposed decision-support tool can solve a real-life, complex industrial problem within a reasonable amount of time, providing important time savings compared to the manual scheduling process. It also offers improvements compared to the manual schedule that mainly focused on feasibility: it helps in reducing the number of temporary workers and in making better use of the qualifications of each employee.

One prospect for future work would be to support the managers in setting their parameters, particularly by automatically tuning these optimization parameters through machine learning methods (see *e. g.* Jurisica [108]).

Note that, although the model was designed for a logistic platform, it could be adapted to other service systems, such as mail processing or cleaning companies. Another opportunity for future research would be to apply this decomposition method to similar problems.

Finally, we provide new real-life instances that can be used to benchmark different scheduling techniques. It would be interesting to compare, for each of those instances, the performance of our method with techniques like column generation, the decomposition methods described by Detienne *et al.* [61] and Naudin *et al.* [154], or constraint programming approaches.

*All we have to decide is what to do
with the time that is given us.*

— J. R. R. Tolkien

Chapter 6

INTEGRATED TRUCK SCHEDULING AND EMPLOYEE ROSTERING

In this last chapter, we show how the truck scheduling model from chapter 2 and the employee timetabling and rostering model from chapter 5 can be combined to address both problems in an integrated manner. Three approaches are compared. The sequential approach consists in sequentially solving the different models at our disposal: first IP* or H2, from which a workload is deduced and used as input for MILP1, MILP2 and MILP3. The iterative approach, inspired by Weide *et al.* [213], consists in solving both problems one after another until a stable point is reached. Two iterative procedures are proposed, employees-first and trucks-first.

PLANIFICATION INTÉGRÉE DES CAMIONS ET DES EMPLOYÉS

Dans ce dernier chapitre, on montre comment combiner le modèle de planification de camions du chapitre 2 d'une part, et le modèle de génération des emplois du temps des employés du chapitre 5 d'autre part, afin de traiter les deux aspects de façon intégrée. On adapte à notre problème une idée proposée par Weide *et al.* [213], qui consiste à résoudre les deux modèles l'un après l'autre de façon itérative, jusqu'à atteindre un point stable. On compare trois approches. L'approche séquentielle est une approche intuitive qui pourrait être utilisée par un manager ; elle consiste à résoudre d'abord IP* ou H2 pour en déduire une charge de travail qui sert de donnée d'entrée à MILP1, MILP2 et MILP3. Deux approches itératives sont ensuite proposées : l'une qui résout d'abord le modèle dédié aux employés, et l'autre qui commence par le modèle de planification des camions. Les contraintes d'IP* d'une part et de MILP3 d'autre part sont légèrement modifiées, afin d'introduire davantage de souplesse pour permettre à chaque modèle d'influencer l'autre, et pour les relier via de nouveaux éléments dans les fonctions objectif. Les résultats numériques présentent une étude exploratoire. On montre sur un exemple en quoi l'approche itérative domine l'approche séquentielle. Les résultats sur une série de petites instances permettent ensuite de montrer que les meilleurs résultats pour l'approche itérative sont obtenus lorsqu'on détermine le planning de camions en premier.

6 INTEGRATED TRUCK SCHEDULING AND EMPLOYEE ROSTERING

Chapter 2 and chapter 4 propose several models to schedule truck and pallet moves in a cross-docking platform. In chapter 5, a sequential approach is used to create weekly timetables and daily rosters for logistic platform employees. Truck-related models and employee-related models have been described independently, they actually are strongly linked: the work planned by the truck scheduling models cannot be carried out without logistics employees.

> "To achieve globally optimal solutions, the interdependencies between the different planning functions should be taken into account, and planning decisions should be made simultaneously. In other words, planning problems should be integrated".
>
> Maravelias and Sung [139]

This chapter demonstrates how the two models can be combined to create an integrated decision-support model for a cross-docking platform.

6.1 PROBLEM DESCRIPTION

The model described in chapter 5 can be applied to any type of logistic platform, and can therefore be used for the special case of a cross-docking platform. In the truck scheduling model described in chapter 2, the internal capacity of the platform is expressed only by a constant M, which is the maximum amount of pallets that can be transferred within one time unit.

The problem consists in combining the two models. The employee timetabling and rostering should be done for the specific case of a cross-docking platform, while truck scheduling should incorporate detailed information on the workforce's availability.

6.1.1 *Assumptions*

All assumptions described in section 2.1.1 on the one hand, and in section 5.1.1 on the other hand, still hold. Only the assumption regarding M is now relaxed in the truck scheduling model: the internal capacity is not necessarily considered as a constant, but can vary throughout the day according to the staffing decisions that have been made.

Six different tasks already listed in chapter 1 (see *e.g.* Figure 1.2 on page 11) are considered in the integrated model:

0. unloading an inbound truck,
1. controlling and scanning the unloaded content,
2. direct transfer (from an inbound truck to an outbound truck),
3. transfer to stock,
4. transfer from stock,
5. loading an outbound truck.

It is assumed that these different tasks require different skills from the employees. For example, carrying out the control and scan requires a training on the different control points, and on the use of the WMS. Direct transfers can be done with a hand pallet truck or a powered pallet truck depending on the size of the platform, while a transfer to or from stock would require a forklift truck (see Figure 1.3 on page 12) – thus different licenses are needed.

For detailed standard time calculations, see Appendix D.

The time needed to carry out each one of the different tasks, noted ST for "standard time", is expressed in hour/pallet. As done already in chapter 3, the values are obtained from the classic crossdock sizes given by Bartholdi and Gue [19], and standard process times for logistic operations (Gauvreau [78]). The values used in this chapter and detailed in Table 6.1 are an average between the worst case and the best case.

	Task	Process time (h/pallet)
ST^0	Unloading	0.0492
ST^1	Control and scan	0.0181
ST^2	Direct transfer	0.0583
ST^3	Transfer to stock	0.0583
ST^4	Transfer from stock	0.0583
ST^5	Loading	0.0492

Table 6.1: Process times for cross-docking operations

6.1.2 *Similar problems in the literature*

As shown in chapter 1, in cross-docking literature resource constraints are not often taken into account, let alone detailed timetabling issues. Only Ko *et al.* [111] integrate "fairness" when solving a truck-to-door assignment problem: the objective is to minimize both the number of workers engaged in loading operation and the imbalance ratio among the workers. They use a genetic algorithm approach with a line balancing heuristic. Li *et al.* [127] are the only ones who attempted a totally integrated approach: they propose an Excel tool (the exact functioning of which is not really provided) to conduct the operations planning, sequencing, real-time scheduling for container arrivals and pallet transfer, and real-time resource management. Although the detailed models are not given in the article, their approach seems to be based on greedy heuristics.

It is necessary to turn to different fields to find examples of combined operations planning and employee timetabling using exact methods: production planning on the one hand, and vehicle and crew scheduling on the other hand. Artigues *et al.* [13] give a review of articles dealing with the integration of task and employee scheduling in both application fields. Since the publication of this review in 2007, more recent work was done on the topic. Artigues *et al.* [14] use a hybrid branch-and-bound to solve an integrated employee timetabling and job-shop scheduling problem. Working on two comparable problems, Guyon *et al.* [97, 98] propose to use a Benders decomposition, a specific decomposition with cut generation, and a hybridization of a cut generation process with a branch and bound strategy. In the transportation field, Mercier and Soumis [148] propose an integrated model for aircraft routing, crew scheduling and flight retiming, solved with a Benders decomposition method. Alternatively, Weide *et al.* [213] propose to solve the two models (aircraft routing and crew scheduling) in an iterative way. Traditionally, the routing problem is solved prior to the crew scheduling problem; but the authors note that this procedure might cause some crews to have a very short amount of time to transfer from one aircraft to another, which is likely to propagate delays. By solving both models in an integrated way, they aim at increasing the overall robustness of the operations.

> "We start with a minimal cost crew pairing solution without taking aircraft routings into account. Then, in each iteration we solve the individual aircraft routing problem first, taking into account the current crew pairing solution. Then, given the aircraft routing solution we resolve the crew pairing problem. We only use the objective functions in both problems to pass information from the problem solved previously to generate more and more robust solutions. [...] We stop the process when the level of robustness cannot be improved any further".
>
> Weide *et al.* [213]

The analysis carried out in chapter 1 highlights the gap between the cross-docking literature and industry needs regarding crossdock employee timetabling and rostering. In this chapter, we propose to apply a procedure comparable to the one used by Weide *et al.* [213] in order to connect the truck scheduling and the employee rostering models introduced in the previous chapters.

6.2 SCHEDULING TRUCKS AND EMPLOYEES TOGETHER

A simple sequential approach, that could be used by a manager having both decision support tools at his disposal, is described in section 6.2.1 in order to have a comparison reference when evaluating the iterative approaches described in section 6.2.2; two different iterative strategies (employees-first and trucks-first) are detailed.

6.2.1 *Sequential approach*

The sequential approach is the "intuitive" one, which could be used by a manager who has at his disposal both the truck scheduling tool described in chapter 2 and the weekly timetabling and daily rostering tool described in chapter 5.

The employee timetabling models needs a workload as input, workload which is directly linked to the truck schedule. Yet the truck schedule is difficult to obtain in a cross-docking platform. Hence it would be natural to first run the truck scheduling model for each day of the week – using IP* for small instances, or H2 for bigger ones, or one of their robust versions proposed in chapter 4. The workload for the week can then be deduced from the truck schedules (see the detailed procedure below) and used as input to run the weekly steps of the timetabling process. The daily roster is created every morning, using the workload deduced from the truck schedule of the day, and the timetable already created for the week. MILP3 thus creates a schedule that matches the workload and does not differ too much from the weekly schedule. The process is summarized in Figure 6.1.

DEDUCING WORKLOAD W^1 FROM THE RESULT OF IP*. Among the outputs of IP* are X_{hio}, which gives the direct moves of pallets from truck i to truck o within time unit h; s^I_{hic} which denotes the

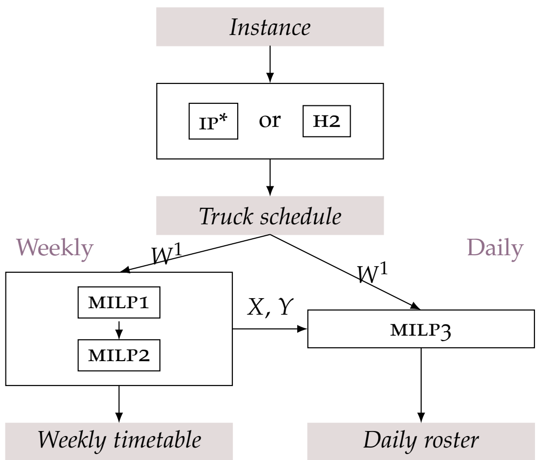


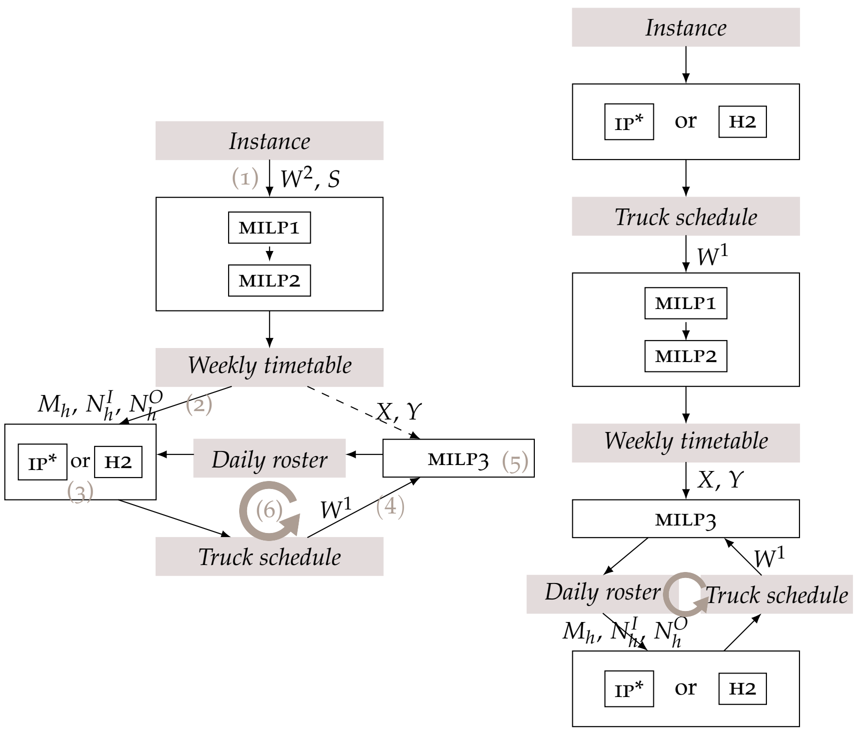
Figure 6.1: Principle of the sequential approach

moves from truck i to storage at time h (for each client c) and s^O_{ho} which gives the number of pallets transferred from storage to truck o at time h. Using these three outputs, the workload can be expressed precisely, hour by hour: all tasks therefore belong to $\mathcal{T}^1$. The workload is defined as follows for all $h \in \mathcal{H}$:

$$\begin{array}{lrcl}
\text{Unloading} & W^1_{0dh} & = & (\sum_{i\in\mathcal{I},o\in\mathcal{O}} x_{hio} + \sum_{i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic}) \times ST^0 \\
\text{Control and scan} & W^1_{1dh} & = & (\sum_{i\in\mathcal{I},o\in\mathcal{O}} x_{hio} + \sum_{i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic}) \times ST^1 \\
\text{Direct transfer} & W^1_{2dh} & = & \sum_{i\in\mathcal{I},o\in\mathcal{O}} x_{hio} \times ST^2 \\
\text{Transfer to stock} & W^1_{3dh} & = & \sum_{i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic} \times ST^3 \\
\text{Transfer from stock} & W^1_{4dh} & = & \sum_{o\in\mathcal{O}} s^O_{ho} \times ST^4 \\
\text{Loading} & W^1_{5dh} & = & (\sum_{o\in\mathcal{O}} s^O_{ho} + \sum_{i\in\mathcal{I},o\in\mathcal{O}} x_{hio}) \times ST^5
\end{array}$$

6.2.2 *Iterative approaches*

The sequential approach described in the previous section does not guarantee global optimality. Although the employee timetable and roster match the previously calculated truck schedule, maybe a better solution could be reached if the trucks schedule was calculated taking staffing issues into account. We therefore apply an approach similar to the one described by Weide *et al.* [213] to our problem. The truck schedule and the employee roster are run iteratively until a sta-



(a) Employees first (b) Trucks first

Figure 6.2: Principle of the iterative approaches

ble point is reached. Two different cases are studied: starting with the calculation of the employee timetable and roster (*employees-first*) and starting with the truck schedule (*trucks-first*). Both principles are described in Figure 6.2 and further detailed in the following sections.

6.2.2.1 *Employees-first*

This solution considers the timescale of the different decisions to be made and therefore calculates first the employees weekly timetabling; the output is communicated to the employees one week in advance. In the following we detail the steps to follow in the employee-first procedure.

Step (1) in Figure 6.2

DEDUCE WORKLOAD W^2 FROM AN INSTANCE. A difficulty of this approach is that the employee timetable has to be calculated before the actual truck schedule is known, since IP* has not been run yet at this stage. Hence, the workload has to be estimated. The proposed solution is to define all tasks as defined by slots, *i. e.* all tasks belong to set $\mathcal{T}^1$. The slots are defined based on the wishes of the transportation providers. MILP1 and MILP2 take the decision about when to carry out the different tasks, within the predefined slots. In order to quantify the workload regarding storage, an estimation τ^{stock} is given as the proportion of pallets which usually go to stock – based *e. g.* on historical data. More precisely, workload W^2 and slots S are defined for day d as follows:

Unloading	$W^2_{0d} =$	$\lvert\mathcal{I}\rvert \times ST^0$
	$S_{0hd} =$	1 if h is in the wished presence time window of an inbound truck;
Control and scan	$W^2_{1d} =$	$\lvert\mathcal{I}\rvert \times ST^1$
	$S_{1hd} =$	1 if h is in the wished presence time window of an inbound truck;
Direct transfer	$W^2_{2d} =$	$(1 - \tau^{\text{stock}}) \times ST^2$
	$S_{2hd} =$	1 if h is in the wished presence time window of an inbound truck;
Transfer to stock	$W^2_{3d} =$	$\tau^{\text{stock}} \times ST^3$
	$S_{3hd} =$	1 if h is in the wished presence time window of an inbound truck;
Transfer from stock	$W^2_{4d} =$	$\tau^{\text{stock}} \times ST^4$
	$S_{1hd} =$	1 if h is in the wished presence time window of an outbound truck;
Loading	$W^2_{5d} =$	$\lvert\mathcal{I}\rvert \times ST^5$
	$S_{5hd} =$	1 if h is in the wished presence time window of an outbound truck.

DEDUCE NEW DATA M_h, N_h^I, N_h^O FROM THE WEEKLY TIMETABLE. The staffing decisions made in the weekly schedule create some constraints for the platforms operations, in terms of the number of persons available to carry out the different tasks. Three new data elements are thus calculated from the weekly timetable:

Step (2) in Figure 6.2

M_h maximum number of pallets that can be transferred at time unit $h \in \mathcal{H}$, according to the weekly employees timetable;

N_h^I maximum number of pallets that can be unloaded at time unit $h \in \mathcal{H}$, according to the employees' weekly schedule;

N_h^O maximum number of pallets that can be loaded at time $h \in \mathcal{H}$, according to the employees' weekly schedule.

The values of M_h, N_h^I and N_h^O are deduced from the allocation of employees to the transfer, unloading and loading tasks ($t = 0$, $t = 2$, $t = 5$). For a given day d, they are calculated from the output of MILP2 x'_{etdh} as follows:

$$M_h = \sum_{e \in \mathcal{E}} \frac{x'_{e2dh}}{ST^2} \qquad \forall h \in \mathcal{H} \quad (6.1)$$

$$N_h^I = \sum_{e \in \mathcal{E}} \frac{x'_{e0dh}}{ST^0} \qquad \forall h \in \mathcal{H} \quad (6.2)$$

$$N_h^O = \sum_{e \in \mathcal{E}} \frac{x'_{e5dh}}{ST^5} \qquad \forall h \in \mathcal{H} \quad (6.3)$$

M_h is obtained from the allocation of employees to task 2 (direct transfer). Variable x'_{etdh}, which gives a number of persons, is divided by the standard time of the operations (in hour/pallet) to obtain a number of pallet for each hour. Similarly, N^I and N^O are calculated from the allocation to tasks 0 (unloading) and 5 (loading), respectively.

Step (3) in Figure 6.2

INCLUDE NEW DATA M_h, N_h^I, N_h^O IN IP* OR H2. The truck daily schedule is calculated every day using IP* for small instances, or H2 for bigger ones. In order to take into account the new staffing-related information as soft constraints, three new constraints are added to IP*:

$$\textstyle\sum_{o \in O, i \in \mathcal{I}} x_{hio} + \sum_{i \in \mathcal{I}, c \in \mathcal{C}} s_{hid}^I \leq N_h^I + \delta_h^I \qquad \forall h \in \mathcal{H} \quad (10.1)$$

$$\textstyle\sum_{o \in O, i \in \mathcal{I}} x_{hio} + \sum_{o \in \mathcal{O}} s_{ho}^O \leq N_h^O + \delta_h^O \qquad \forall h \in \mathcal{H} \quad (10.2)$$

$$\textstyle\sum_{o \in O, i \in \mathcal{I}} x_{hio} \leq M_h + \varepsilon_h \qquad \forall h \in \mathcal{H} \quad (10.3)$$

$$\Pi_0^\delta = \textstyle\sum_{h \in \mathcal{H}} \delta_h^I + \delta_h^O \quad (53)$$

$$\Pi_0^\varepsilon = \textstyle\sum_{h \in \mathcal{H}} \varepsilon_h \quad (54)$$

Constraint sets (10.1), (10.2) and (10.3) give a penalty point each time the soft constraint is violated. The sums of these penalty points, defined by constraints (53) and (54), are then added to the objective function, thus the new objective is to minimize $\alpha_0 \Pi_0^\alpha + \beta_0 \Pi_0^\beta + \gamma_0 \Pi_0^\gamma + \delta_0 \Pi_0^\delta + \varepsilon_0 \Pi_0^\varepsilon$.

Step (4) in Figure 6.2

DEDUCE W FROM THE RESULT OF IP*. After IP* is solved with the new constraints and new objective function, the output is used to calculate workload W as detailed in section 6.2.1. The workload is used as an input in MILP3, together with the values of X and Y fixed by MILP2.

Step (5) in Figure 6.2

ADD INTERVAL FLEXIBILITY IN MILP3. For the daily truck schedule and employee roster to be able to influence each other until a stable point is reached, it is important to leave some flexibility to MILP3 regarding the intervals in which the work can be done. Therefore, constraint set (24) defined in section 5.2.2:

$$\textstyle\sum_{e\in\mathcal{E}''} x''_{et_1q} = W^1_{t_1q} \qquad \forall t \in \mathcal{T}^1, q \in \mathcal{Q} \quad (24)$$

is replaced by constraint sets (24.1), (24.2) and (24.3) as follows:

$$\textstyle\sum_{e\in\mathcal{E}''} x''_{et_1q} = W^1_{t_1q} + \varepsilon^+_{t_1q} - \varepsilon^-_{t_1q} \qquad \forall t \in \mathcal{T}^1, q \in \mathcal{Q} \quad (24.1)$$

$$\textstyle\sum_{e\in\mathcal{E}'',q\in\mathcal{Q}} x''_{et_1q} = \sum_{q\in\mathcal{Q}} W^1_{t_1q} \qquad \forall t \in \mathcal{T}^1 \quad (24.2)$$

$$\textstyle\Pi^\varepsilon_3 = \sum_{t\in\mathcal{T}^1,q\in\mathcal{Q}} \varepsilon^+_{t_1q} + \varepsilon^-_{t_1q} \quad (24.3)$$

Constraint set (24.1) replaces constraint set (24) and changes it into a set of soft constraints. Constraint (24.2) ensures that, despite the flexibility provided to replace the work in different time slots, the total amount of hours worked still matches the workload. The objective function is changed in order to add Π^ε_3, defined in constraint (24.3), to the objective function of MILP3.

Step (6) in Figure 6.2

ITERATE UNTIL REACHING A STABLE POINT. Using the daily roster output of MILP3, the values of M_h, N^I_h and N^O_h can be updated and used to run IP* again. The new versions of IP* and MILP3 are run iteratively until a stable point is reached. The stable point is considered reached when the values of the different penalties that measure adjustments, *i.e.* Π^δ_0, Π^ε_0, Π^α_3, Π^β_3 and Π^ε_3, are stable. Table 6.2 gives a reminder of the different penalties described in section 2.2.1 for IP* and section 5.2.2 for MILP3. In some cases, the iteration does not converge to a single stable point but to a set of two, three or more solutions (oscillator): in this case the loop is stopped and the solution with the smallest objective function Π_3 is chosen.

	IP* *penalties*		MILP3 *penalties*
Π^α_0	inbound truck time window penalty	Π^α_3	shift changes
Π^β_0	outbound truck time window penalty	Π^β_3	task changes
Π^γ_0	number of pallets in storage	Π^γ_3	handling equipment penalty
Π^δ_0	transfer capacity violations	Π^δ_3	ergonomy penalty
Π^ε_0	loading/unloading capacity violations	Π^ε_3	interval changes for tasks in $\mathcal{T}^1$

Table 6.2: IP* and MILP3 penalties description (reminder)

6.2.2.2 *Trucks-first*

Calculating the employees timetable first can favor the employees, but leaving the employees-related MILPs to decide when the trucks should be docked could lead to strongly sub-optimal truck schedules. In order to prevent that problem, the trucks-first approach starts as the sequential approach: IP* or H2 is used to calculate a truck schedule from the instance. The workload W is calculated from the truck schedule (see section 6.2.1 for details) and used as input to generate the weekly schedule, followed by the daily roster. While the sequential approach stops there, the iterative approach questions this daily roster to adapt it to the truck schedule constraints.

From the output x'' of the daily roster, one can calculate the values of M_q, N_q^I and N_q^O, which are capacity constraints at time interval $q \in \mathcal{Q}$ for the transfer, unloading and loading operations, respectively. The values of these data elements are calculated as detailed in section 6.2.2.1. The truck schedule is then obtained with the new version of IP* or H2 described in section 6.2.2.1, with constraints sets (10.1), (10.2) (10.3), (53) and (54). Based on this truck schedule, a new workload W is calculated and used as input for MILP3 as well as the outputs of MILP2 X an Y. The version of MILP3 used also replaces constraint set (24) by constraints sets (24.1), (24.2) and (24.3) as detailed in section 6.2.2.1, in order to add flexibility regarding the possible intervals to execute each task.

Similar to the employees-first approach, IP* or H2 and MILP3 are run iteratively until a stable point or an oscillator is reached – for the latter, the solution with the smallest objective function Π_0 is chosen.

6.3 NUMERICAL RESULTS

In this section, exploratory numerical experiments are carried out: the aim is to demonstrate that the method detailed in section 6.2.2 is a valid way to combine the truck scheduling model with the employee scheduling model. After a presentation of the instance generation process in section 6.3.1, section 6.3.2 uses an example to show how the iterative approach outperforms the sequential approach. In section 6.3.3, the performances of both iterative approaches (truck-first and employees-first) are compared and discussed.

6.3.1 *Instance generation*

The truck-related parts of the instances correspond to the instance `set3+3` described in section 2.2.3. The employee-related parts of the instances are generated randomly, using the principle detailed in section 5.3.3, with the number of employees set to 10 for the instances where $M = 17$, and set to 15 for the instances where $M = 34$. The time horizon (number of hours $|\mathcal{H}|$) on the employees side is set equal to the value of $|\mathcal{H}|$ on the trucks side.

In order to keep the weekly and daily stages easily comparable, the time unit considered when creating the daily roster (interval) has a length of one hour, thus $\mathcal{H} = \mathcal{Q}$.

The value of τ^{stock}, estimation of the percentage of pallets that go through storage, is set to 3%. The handling equipment upper bound Max_{tq} and the safety and ergonomics bound Max_t are set to ∞ for all $t \in \mathcal{T}, q \in \mathcal{Q}$ so that Π_3^{γ} and Π_3^{δ} will always be 0.

6.3.2 *Comparison sequential / iterative approaches*

When introducing the iterative approach, we pointed out the fact that reaching a local optimum for both models separately does not necessarily mean reaching a good solution when both are combined. This point is illustrated in this section by applying the sequential procedure and an iterative one (here trucks-first) to instance 17_1. Recall from section 2.2.3 that instance 17_1 has a time horizon $|\mathcal{H}| = 10$, 5 inbound and 5 outbound trucks ($|\mathcal{I}| = |\mathcal{O}| = 5$) serving 3 different clients.

SEQUENTIAL APPROACH. For a small instance like 17_1, the first step of the sequential approach as described in section 6.2.1 is to run IP*. The solution obtained, with an objective value of 0 ($\Pi_0^{\alpha} = 0$, $\Pi_0^{\beta} = 0$, $\Pi_0^{\gamma} = 0$) was already displayed in Figure 2.8 on page 53. As a first approximation, let us assume that this truck schedule will apply to each of the five days of the week. The workload W corresponding to this truck schedule, as well as the qualification matrix Q used in this instance, are as follows:

$$W_{td} = \begin{bmatrix} 8 & 3 & 10 & 0 & 0 & 8 \\ 8 & 3 & 10 & 0 & 0 & 8 \\ 8 & 3 & 10 & 0 & 0 & 8 \\ 8 & 3 & 10 & 0 & 0 & 8 \\ 8 & 3 & 10 & 0 & 0 & 8 \end{bmatrix} \qquad Q_{et} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Note that two tasks related to storage have a null workload, since no pallet is put in storage in this solution. Using this workload as an input, MILP1 and MILP2 give the weekly timetable shown in Figure 6.3.

Employees 0, 2 and 5 are not put to work in this timetable and are absent all week. Running MILP3 for day $d = 0$ (Monday) gives a daily roster exactly equal to the one displayed in Figure 6.3 for Monday,

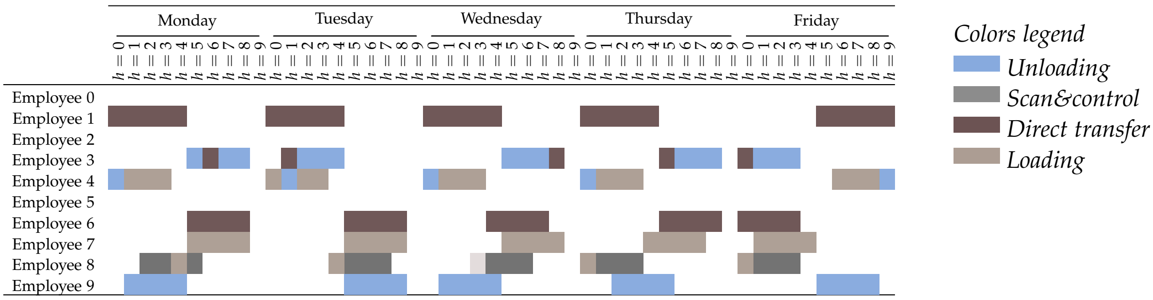


Figure 6.3: Sequential approach on 17_1: weekly timetable

thus the objective value for MILP3 is 0 ($\Pi_3^\alpha = 0$, $\Pi_3^\beta = 0$, $\Pi_3^\gamma = 0$ and $\Pi_3^\delta = 0$).

When looking at the objective functions only, this approach seems very good since each model, taken independently, is solved to optimum with no soft constraint violated. But can these two results (truck timetable and employee daily roster) be combined easily? Looking at the number of employees allocated to each task at the different time units on Monday (Figure 6.3), and using equations 6.1 to 6.3, we can calculate the employee capacities available at every time unit $h \in \mathcal{H}$:

$$\begin{array}{lcl} M & = & [\ 17\ \ 17\ \ 17\ \ 17\ \ 17\ \ 17\ \ 34\ \ 17\ \ 17\ \ 0\ ] \\ N^I & = & [\ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 0\ \ 0\ \ 20\ \ 20\ \ 0\ ] \\ N^O & = & [\ 0\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 0\ ] \end{array}$$

For example, Figure 6.3 shows that two employees are allocated to direct transfer at time $h = 6$, therefore $M_6 = 34$.

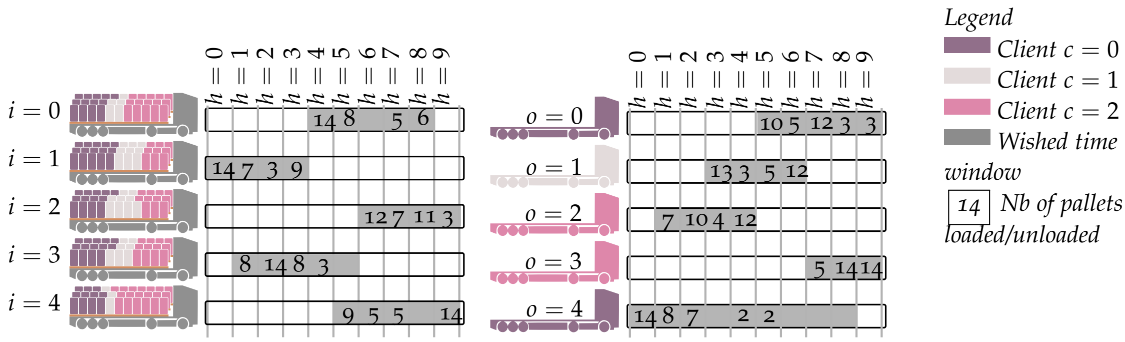


Figure 6.4: Sequential approach on 17_1: truck schedule

Looking at the truck schedule used by the sequential approach (displayed in Figure 2.8 on page 53 and presented in a more compact form in Figure 6.4) we can see that those capacity constraints are violated many times. The loading/unloading capacities N^I and N^O are violated for 51 pallets in total (all the pallets loaded or unloaded when the capacity is 0 for those tasks), and the transfer capacity M for 17 pallets (all the pallets transferred at time $h = 9$). That would be equivalent to objective values $\Pi_0^\delta = 51$ and $\Pi_0^\varepsilon = 17$. Is it possible to do better with the trucks-first approach?

TRUCKS-FIRST APPROACH. The trucks-first approach starts exactly like the sequential approach, but the values of M, N^I and N^O are now integrated to IP* as soft constraints. The result, displayed in Figure 6.5, yields to the objective function $\Pi_0 = 58$ where $\Pi_0^\alpha = 1$, $\Pi_0^\beta = 0$, $\Pi_0^\gamma = 0$, $\Pi_0^\delta = 57$, $\Pi_0^\varepsilon = 0$.

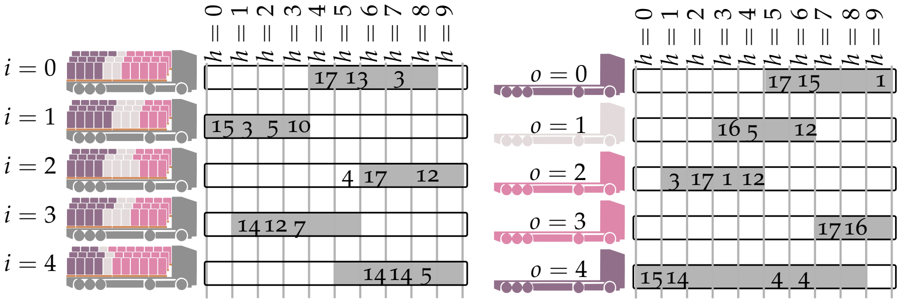


Figure 6.5: Iteration 1 on 17_1: truck schedule

From this truck schedule, the detailed workload for each interval $q \in \mathcal{Q}$ can be expressed as shown on this page. Then, using this workload as input, MILP3 is run again to give the result shown in Figure 6.6. The corresponding penalties are $\Pi_3^\alpha = 2$, $\Pi_3^\beta = 6$, $\Pi_3^\gamma = 0$, $\Pi_3^\delta = 0$, $\Pi_3^\varepsilon = 8$. It means that there is a 2-hour change in the allocated shifts compared to the result of MILP2 (for employee 9) and 6 hours of task changes (for the tasks of employee 8).

$$W_{t_1 q} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

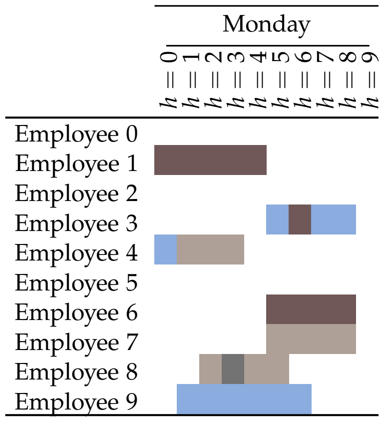


Figure 6.6: Iteration 2 on 17_1: employee roster

New capacity constraints can be derived from this employee roster:

$$M = [\ 17\ \ 17\ \ 17\ \ 17\ \ 17\ \ 17\ \ 34\ \ 17\ \ 17\ \ 0\ ]$$
$$N^I = [\ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 40\ \ 20\ \ 20\ \ 20\ \ 0\ ]$$
$$N^O = [\ 0\ \ 20\ \ 40\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 0\ ]$$

and used in IP* to obtain the truck schedule in Figure 6.7, with penalties $\Pi_0^\alpha = 1$ (because of the hour added at $h = 5$ for inbound truck $i = 2$), $\Pi_0^\beta = 0$, $\Pi_0^\gamma = 0$, $\Pi_0^\delta = 37$, $\Pi_0^\varepsilon = 0$.

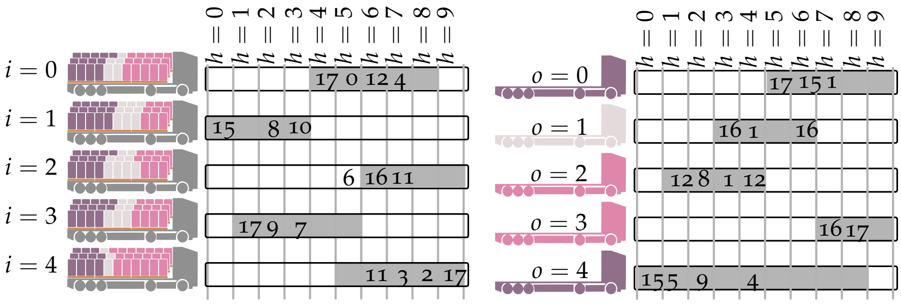


Figure 6.7: Iteration 2 on 17_1: truck schedule

The next iteration yields exactly the same solution – therefore the procedure stops after three iterations in total. The comparison between the sequential and the truck first approach, in terms of value of the objective function, is done in Table 6.3. The trucks-first approach reduces the values of Π_0^δ and Π_0^ε, *i. e.* reduces the violations of the staff-related capacity constraints. It also increases the value of Π_0^α (one inbound truck is assigned to a time windows slightly different from its wish) and the difference between the weekly timetable and the daily roster (Π_3^α, Π_3^β), but it is a price to pay to make the truck schedule and the employee roster more compatible.

	IP*					MILP3				
	Π_0^α	Π_0^β	Π_0^γ	Π_0^δ	Π_0^ε	Π_3^α	Π_3^β	Π_3^γ	Π_3^δ	Π_3^ε
Sequential	0	0	0	51	17	0	0	0	0	0
Trucks-first	1	0	0	37	0	2	6	0	0	8

Table 6.3: Sequential/trucks-first results for 17_1

6.3.3 *Comparison employees-first / trucks-first*

Intuitively, one could think that the employees-first procedure favors the employees' wishes, while the truck-first procedure favors the transportation providers' wishes instead. The results obtained on instance `set3+3`, displayed in Table 6.4, confirm this idea. Most of the time, the values of Π_3^α and Π_3^β are smaller for the employees-first approach. All the other penalties, however, are bigger for the employee approach. The penalties regarding truck time windows assignments (Π_0^α and Π_0^β), especially, are significantly bigger for the employees-first approach compared to the trucks-first approach. On this set of small instances, the trucks-first approach therefore dominates the employees-first approach.

	Employees-first											Trucks-first										
	Ite.	Π_0^α	Π_0^β	Π_0^γ	Π_0^δ	Π_0^ε	Π_3^α	Π_3^β	Π_3^γ	Π_3^δ	Π_3^ε	Ite.	Π_0^α	Π_0^β	Π_0^γ	Π_0^δ	Π_0^ε	Π_3^α	Π_3^β	Π_3^γ	Π_3^δ	Π_3^ε
17_1	3	1	1	0	135	1	2	3	0	0	14	3	1	0	0	37	0	2	6	0	0	8
17_2	4	8	12	0	195	0	2	3	0	0	14	3	0	1	0	31	0	3	6	0	0	6
17_3	5	16	16	0	208	0	2	2	0	0	24	3	0	2	0	31	0	3	11	0	0	6
17_4	2	15	15	0	208	0	0	0	0	0	24	3	0	0	0	51	1	3	9	0	0	4
17_5	3	14	12	0	198	6	2	3	0	0	22	3	0	3	0	51	0	3	9	0	0	4
34_1	3	0	0	0	52	0	4	3	0	0	6	3	0	0	0	38	0	4	4	0	0	4
34_2	3	0	0	0	72	8	4	4	0	0	14	4	0	1	0	42	0	4	8	0	0	4
34_3	2	0	1	0	115	14	2	1	0	0	14	3	0	1	0	44	0	0	0	0	0	4
34_4	3	12	12	0	210	0	2	4	0	0	22	5	1	2	0	51	0	2	8	0	0	6
34_5	4	18	20	0	210	12	2	4	0	0	24	5	0	2	0	51	1	2	9	0	0	6
34_6	5	24	24	0	198	0	4	4	0	0	24	5	0	3	0	48	0	6	8	0	0	6

Table 6.4: Results for both iterative approaches

6.4 CONCLUSION

This chapter demonstrates how an iterative procedure can be used to combine the models described in chapter 2 and chapter 5, namely a truck scheduling model on the one hand and an employee weekly timetabling and daily rostering problem on the other hand. Numerical experiments on small instances show that the best results are obtained when the truck scheduling model is run first. Further work is needed to check whether this result scales-up for bigger instances, and to analyze the behavior of the system when the different parameters change.

This chapter uses IP* as truck scheduling model: an extension of this work would be to use instead some of the robust versions described in chapter 4. The simulation model described in chapter 3 could also be adapted to properly model the human resources of the platform, and therefore used to evaluate the robustness of the integrated timetable and roster.

The limits of this approach reside in the fact that no fully integrated model is available, therefore the quality of the solutions given by the iterative process cannot be compared to the optimal value. A model integrating all the industrial constraints of the truck scheduling and the employee rostering would probably be too hard to be solved. However, the different decomposition processes proposed by Guyon *et al.* [97, 98] might be applicable to our case (or a simplified version of it). They are exact methods yielding to optimal solutions. Specifically, the cut generation process presented in [97] splits the model into a master problem, which assigns a work pattern to each operator, and a sub-problem which checks the feasibility of the assignment – this sub-problem is actually a maximum flow problem on a directed transportation network. Because the crossdock truck scheduling problem also contains a maximum flow problem as a subproblem (see section 2.3.3), applying the method proposed by Guyon *et al.* to the cross-docking environment seems a promising idea. Applying Benders decomposition is also a possible perspective in order to get an exact solution to the integrated problem.

Science never solves a problem
without creating ten more.

— George Bernard Shaw

CONCLUSION AND PERSPECTIVES

CONCLUSION AND PERSPECTIVES

SUMMARY OF THE CONTRIBUTIONS

In a decade of highly tensed economical context but also very fast progress in new technologies, industries have to undertake a mutation to adapt themselves. It becomes critical for them to have a fast, efficient and reactive supply-chain.

Cross-docking is an example of Just-In-Time technique in logistics. By transferring products from inbound trucks to outbound trucks with almost no temporary storage, it speeds up the delivery flow while reducing inventory costs. However, like any Just-In-Time process, a cross-docking platform needs a flawless scheduling system to operate properly.

This dissertation focuses on the operations within a cross-docking platform. As a first contribution, chapter 1 establishes a picture of the state-of-the-art in the cross-docking literature on the one hand, and of the reality encountered daily by crossdock managers on the other hand. Besides proposing unified problem names and a comparison framework that can be reused by experts in the field, this study brings out two major gaps between the current state of research and the practice in industry. The first issue is a concern about truck punctuality, that is not often taken into account in the cross-docking models of the literature; the second issue is the scheduling of human resources, which are crucial in a crossdock as its first cost center.

Chapter 2, chapter 3 and chapter 4 address the first issue. In chapter 2, a crossdock truck scheduling model is proposed which takes into account the wishes of the transportation providers regarding their arrival and departure times. The problem, formulated as an Integer Program, is shown to be NP-hard in the strong sense, and three different heuristics are proposed to solve it for rather large instances. Two of the three heuristics are based on a decomposition of the original IP model into IP submodels solved sequentially; the third one is a tabu search in which the objective function is evaluated using a maximum flow graph. In order to address the platform managers' concern about delayed trucks, chapter 3 proposes a methodology to evaluate the robustness of a schedule, *i.e.* its ability to react with as few perturbations as possible to unexpected events. A simulation model is developed to represent the platform operations with various sources of uncertainty. We propose a novel way to combine a simulation model with an optimization model, here the model presented in chapter 2. Robustness metrics are proposed based on the results of the simulation. These metrics are used in chapter 4 to compare different robust reformulations of the original truck scheduling problem. Besides applying standard robust optimization techniques, this chapter proposes to adapt ideas from robust project scheduling and shows that they can perform well to obtain robust crossdock truck schedules.

The second aspect identified as a key issue for crossdock managers is employee timetabling and rostering. Chapter 5 proposes a decomposition of this problem into three sub-problems, corresponding to the three levels of decisions to be made. Each step is modeled with a Mixed and Integer Linear Program and shown to be NP-hard in the strong sense. However, the decomposition enables one to solve instances of realistic sizes in a reasonable amount of time, as evidenced by a successful implementation of our timetabling tool in industry. Different graphical interfaces have been developed for different uses of the tool, one of them for teaching purposes.

Finally, chapter 6 demonstrates how the two models developed independently, namely the truck scheduling model and the employee timetabling and rostering model, can be combined to solve the integrated problem.

Along the entire dissertation, we followed a typical operations research process as illustrated in Figure 7.1. Through visits and interviews in cross-docking platforms, we could identify the cross-docking optimization problems that are relevant for today's industry, and thus write business-specific models. We propose to solve in an integrated manner the operational problems that were identified as key elements. Lastly, the suitability of the model in a business environment is validated by using a simulation model integrating uncertainty.

Figure 7.1: A typical operations research process

PERSPECTIVES

The perspectives for this work can be seen at different levels. Short-term and technical perspectives regarding possible extensions of the models and numerical experiments presented are detailed in the conclusion of each chapter (chapter 2 on page 65; chapter 3 on page 87; chapter 4 on page 111; chapter 5 on page 144; chapter 6 on page 162).

A mid-term perspective could be to combine this work on cross-dock operations scheduling with crossdock network scheduling. Because this dissertation focuses on internal operations, all the network-related problems were set aside. Designing a cross-docking network, or a distribution network including one crossdock or more, is a strategic decision. However, operating this network and scheduling the daily transfers between the different actors of the supply-chain correspond to operational decisions. It is clear that this problem is strongly linked to decisions made by the platform management: if a single truck is used for different transfers in the network, a delay at one platform will impact the whole network. This year, Agustina *et al.* [4] and Dondo and Cerdá [63] started addressing the integrated problem, which needs to be further explored.

Another perspective would be to integrate environmental issues in our model, especially because logistic companies will probably experience a stronger economical pressure on these aspects in the upcoming years. In logistics, a compromise has to be found between the speed of delivery and the truck filling. In a crossdock, what should be done with a truck which is only half-full at the time when it is supposed to leave? Keeping the truck longer impacts the delivery time and the service level; but sending a truck half-full is also a bad decision from the economical and ecological points of view. In this dissertation the assumption was made that all trucks leave full; a detailed study on how to manage this trade-off is a possible extension.

The different models presented in this dissertation (especially the truck-related models) thus require further work before they can be adopted in an industrial context. More realistic assumptions need to be added and the execution times should be shortened for instances representing large platforms.

Another important aspect is the real-time control of logistics operations. The WMS (Warehouse Management Systems) already gather a lot of data in real-time, and this trend is likely to increase in the next years with the development of vocal and augmented-reality technologies. Optimization models should then be run not only on a weekly or daily scale, but also regularly through the day, exploiting the new information to give real-time decision support. It means developing very reactive and fast optimization models, likely to use important amounts of data as inputs to recalculate new schedules.

Once this work is done, the next step will be to integrate the tool in the information system of a logistics platform. Several benchmarks of existing WMS are available – see *e.g.* Supply Chain Magazine [190] for index cards on the solutions used by platforms in France. However it is difficult to have a clear view of how much "optimization" these support systems are using, and which sort of optimization – greedy allocation, local search...? A detailed study would be needed to have a clear picture; but from our experience in industry it seems that optimization techniques are very rarely used in logistic platforms. Integrating our optimization tool into existing WMS in order to use their data as input would thus be the next mid-term step.

The long-term perspectives are strongly linked to the way the logistics industry will evolve in the next decade.

> "Like plants, warehouses belong to a larger supply-chain scheme, and although their intrinsic performance is important, what will make a difference is the way they will be relevantly used".
>
> Freely translated from Polge [160]

Currently emerging trends, which should become the norm by the year 2025 or 2050, are identified in several studies such as those by Deutsche Post DHL [62] or Gue *et al.* [90]. Of course not all the identified trends are linked to cross-docking; but cross-docking platforms and operations scheduling have their part to play in several of them.

Urban logistics and the last-kilometer issue currently form a very active stream of research and a serious challenge. In order to reduce city congestion, new modes of urban freight transportation should be developed. But to keep a fast flow of goods streaming in the city, cross-docking platforms are needed to transfer items from long-haul trucks to those new urban means of transportation.

> "Open shared-use, crossdock facilities that can be dynamically scheduled for use by multiple, often competing retailers may enable more cost-effective last-mile distribution. To create such facilities, equitable and efficient space, door and labor allocation and scheduling systems need to be created".
>
> Gue *et al.* [90]

Our work fits very well in this big picture drawn by Gue *et al.* Applying it to platforms especially dedicated to urban logistics is a possible long-term perspective.

physicalinternet initiative.org

Researchers in Canada, Europe, and the United States have recently proposed a new concept in logistics, called the Physical Internet (PI or π). It is an ambitious vision aiming at transforming the way physical objects are moved, stored, supplied and used – inspired from the way computers are interconnected through Internet. The goal is to entirely reorganize the distribution of goods in an efficient way which is economically, environmentally and socially sustainable. The goods, encapsulated in standardized π-containers, could be handled by dedicated material handling equipment – π-movers, π-conveyors. π-crossdocks or transit centers, possibly multimodal, would then facilitate the truck-to-truck transshipment in a fully automated and very fast way. Of course the model developed in chapter 5 for logistics employees cannot apply in this case: thanks to the high standardization of the π-containers, all transshipment operations could be 100% automated. However, crossdock truck scheduling models would be highly needed to operate smoothly such automated facilities.

CONCLUSION ET PERSPECTIVES

RÉSUMÉ DES CONTRIBUTIONS

Dans un contexte économique tendu où les nouvelles technologies progressent à toute vitesse, les industries doivent évoluer. Posséder une supply-chain rapide, efficiente et réactive est devenu un enjeu critique.

Le cross-docking est un exemple de technique de "juste-à-temps" en logistique. En transférant des produits des camions entrants aux camions sortants avec peu ou pas de stockage intermédiaire, cette technique permet d'accélérer les flux tout en réduisant les coûts de stockage. Néanmoins, comme tout processus en juste-à-temps, une plateforme de cross-docking nécessite un système de planification sans failles pour pouvoir fonctionner correctement.

Cette thèse traite de la gestion des opérations au sein d'une plateforme de cross-docking. Le chapitre 1 est une première contribution qui propose un état des lieux de la littérature en cross-docking d'une part, et de la réalité du quotidien des managers de plateforme d'autre part. Outre la proposition de renommer de façon unifiée les problèmes rencontrés, et la création d'une grille de comparaison qui peut être réutilisée par les experts du domaine, ce travail dégage deux écarts principaux entre l'état actuel de la recherche et les pratiques de l'industrie. Le premier axe concerne la ponctualité des camions, peu prise en compte dans la littérature du cross-docking ; le second axe est la planification des ressources humaines, point crucial dans les plateformes puisqu'elles sont le premier centre de coût.

Les chapitres 2, 3 et 4 développent le premier axe de travail. Dans le chapitre 2, un modèle de planification des camions est proposé qui prend en compte les souhaits des transporteurs concernant leurs heures d'arrivée et de départ. On montre que le problème, formulé comme un programme linéaire en nombres entiers (PLNE), est NP-difficile au sens fort. Trois heuristiques sont proposées pour permettre de le résoudre pour des instances d'assez grande taille. Les deux premières sont basées sur une décomposition du PLNE initial en deux sous-problèmes, modélisés en PLNE et résolus de façon séquentielle. La troisième est une recherche tabou dont la fonction objectif est évaluée par un flot maximum dans un graphe. Afin de répondre à la problématique des managers concernant la gestion des camions en retard, le chapitre 3 propose une méthodologie d'évaluation de la robustesse d'un planning, c'est-à-dire sa capacité à réagir à des événements imprévus avec le moins de perturbations possibles. Un modèle de simulation est développé pour représenter les opérations d'une plateforme soumise à diverses sources d'incertitudes. Ce modèle de simulation est combiné de manière innovante avec le modèle d'optimisation du chapitre 2. Des indicateurs de robustesse sont proposés à partir des résultats de la simulation. Ces indicateurs sont utilisés dans le chapitre 4 pour comparer différentes reformulations robustes du problème initial. En plus des techniques génériques d'optimisation robuste, ce chapitre propose d'adapter des idées provenant de la planification de projets robustes. On montre qu'elles donnent de bons résultats et permettent d'obtenir des plannings robustes pour les camions de la plateforme.

L'élaboration des emplois du temps des employés est le second axe de travail identifié comme un levier important pour les managers de plateforme. Le chapitre 5 propose une décomposition de ce problème en trois sous-problèmes résolus séquentiellement, qui correspondent à trois niveaux différents de décision. Chacune des étapes, modélisée par un programme linéaire mixte, est NP-difficile au sens fort. La décomposition permet cependant de résoudre des instances de taille réaliste dans des délais raisonnables, comme le prouve la mise en œuvre réussie de notre outil de génération d'emplois du temps en industrie. Différentes interfaces graphiques ont été développées pour différents usages de l'outil, dont une destinée spécifiquement à l'enseignement.

Enfin le chapitre 6 montre comment les deux modèles développés indépendamment – le modèle de planification des camions et celui permettant de générer les emplois du temps des employés – peuvent être combinés afin de résoudre le problème intégré.

Au cours de cette thèse, nous avons suivi une démarche typique de recherche opérationnelle, comme illustré en figure 7.2. À la suite de visites et d'entretiens dans des plateformes de cross-docking, nous avons pu identifier les problématiques d'optimisation en cross-docking qui sont pertinentes pour l'industrie actuelle, et ainsi proposer des modèles orientés métier. Nous proposons de traiter de façon intégrée les décisions opérationnelles qui ont été identifiées comme des points clés. Enfin, la pertinence du modèle au niveau métier est validée par un modèle de simulation qui permet d'intégrer l'incertain.

Figure 7.2: Une démarche de recherche opérationnelle

PERSPECTIVES

Les perspectives de ces travaux peuvent être considérées à plusieurs niveaux. Les perspectives techniques et de court terme concernant les extensions possibles des modèles et des résultats numériques présentés sont détaillées dans la conclusion de chacun des chapitres (conclusion du chapitre 2 à la page 65 ; chapitre 3 page 87 ; chapitre 4 page 111 ; chapitre 5 page 144 ; chapitre 6 page 162).

Une perspective à moyen terme est de combiner ce travail sur la planification des opérations de cross-docking avec des problèmes de planification de réseaux de crossdocks. Les problèmes relatifs aux réseaux n'ont pas été abordés dans cette thèse qui ne traite que des opérations internes. La conception de réseaux de crossdocks, ou de réseaux logistiques comprenant un crossdock ou plus, est une décision stratégique. Cependant, la gestion quotidienne de ce réseau et des transferts entre les différents acteurs de la supply-chain est bien un ensemble de décisions opérationnelles. Il est clair que ce problème est intimement lié aux décisions prises par les managers de plateforme : si un seul camion est utilisé pour plusieurs transferts au sein du réseau, un retard sur une plateforme risque d'impacter l'ensemble du réseau. Cette année, Agustina *et al.* [4] et Dondo and Cerdá [63] ont commencé à traiter ces deux problèmes de façon intégrée, idée qui mérite d'être explorée davantage.

Une autre perspective serait de prendre en compte des aspects environnementaux dans notre modèle, en particulier parce que les entreprises logistiques devraient subir une forte pression économique sur ces questions dans les années qui viennent. En logistique, un compromis doit être trouvé entre la vitesse de livraison et le remplissage des camions. Dans un crossdock, que faire avec un camion qui n'est qu'à moitié plein à l'heure planifiée pour son départ ? Garder le camion à quai plus longtemps impacte le délai de livraison et la qualité de service ; mais envoyer sur la route un camion à moitié plein est également une mauvaise décision du point de vue économique comme écologique. Dans cette thèse nous avons postulé que tous les camions partent pleins ; une étude détaillée de la façon de gérer ce compromis est une extension possible.

Les différents modèles présentés dans cette thèse, et notamment les problèmes de planification de camions, nécessitent donc davantage de travail avant de pouvoir être adoptés dans un contexte industriel. Des hypothèses plus réalistes doivent être ajoutées et les temps d'exécution doivent être raccourcis pour traiter rapidement des instances représentant de grandes plateformes.

Un autre aspect important est le contrôle en temps réel des opérations logistiques. Les WMS (logiciels de gestion d'entrepôts) regroupent déjà une grande quantité de données, et cette tendance devrait encore augmenter dans les prochaines années avec le développement des technologies vocales et de réalité augmentée. Les modèles d'optimisation devraient donc être utilisés non plus à l'échelle de la semaine ou de la journée, mais régulièrement dans la journée, exploitant les nouvelles informations pour apporter une aide à la décision en temps réel. Cela implique de développer des modèles d'optimisation très réactifs et rapides, capables d'exploiter des quantités importantes de données pour recalculer de nouveaux plannings.

Une fois ce travail réalisé, l'étape suivante est d'intégrer l'outil au système d'information d'une plateforme logistique. Plusieurs bancs d'essais sont disponibles qui comparent les WMS existants – voir par exemple les fiches proposées par Supply Chain Magazine [190] concernant les solutions utilisées par les plateformes françaises. Il est cependant difficile d'avoir une vision claire sur la part d'"optimisation" utilisée par les logiciels, et sur le type d'optimisation réalisée – allocation gloutonne, recherche locale... ? Une étude détaillée serait nécessaire pour obtenir un panorama clair ; mais notre expérience en industrie suggère que les techniques d'optimisation ne sont pratiquement jamais utilisées par les plateformes logistiques. Intégrer notre outil d'optimisation dans un WMS existant de façon à utiliser les données du logiciel en entrée serait donc l'étape suivante de ce plan à moyen terme.

Les perspectives à long terme sont fortement liées à la façon dont l'industrie de la logistique va évoluer au cours des dix prochaines années.

> "Tout comme les usines, les entrepôts s'inscrivent dans un schéma supply chain plus vaste, et si leur performance intrinsèque est importante, c'est davantage la manière dont ils seront utilisés avec pertinence qui fera la différence".
>
> Polge [160]

Les tendances qui émergent actuellement, et qui pourraient devenir la norme d'ici 2025 ou 2050, sont identifiées dans plusieurs études dont celles menées par Deutsche Post DHL [62] ou Gue *et al.* [90]. Bien entendu, toutes les tendances identifiées ne sont pas liées au cross-docking ; mais les plateformes de cross-docking et la planification de leurs opérations ont leur rôle à jouer dans plusieurs d'entre elles.

La logistique urbaine et la problématique du dernier kilomètre sont un sujet de recherche très actif et un défi majeur. Afin de réduire la congestion dans les villes, de nouveaux modes de transport de fret en ville doivent être développés. Mais afin de conserver un flux rapide de marchandises vers la ville, des plateformes de cross-docking sont nécessaires pour transférer les produits depuis les poids lourds vers ces nouveaux moyens de transport urbain.

> "Des crossdocks en accès ouvert et partagé, capables de planifier de façon dynamique un usage simultané par des distributeurs multiples et souvent concurrents, devraient

> permettre une distribution rentable pour le dernier kilomètre. Afin de mettre en place de telles installations, il est nécessaire de créer des systèmes équitables et efficaces pour l'allocation et la planification de l'espace, des portes et de la main d'œuvre".
>
> Traduit librement de Gue *et al.* [90]

Notre travail s'intègre bien dans cette vision proposée par Gue *et al.* L'appliquer à des plateformes spécialement dédiées à la logistique urbaine constitue une possible perspective à long terme.

Des chercheurs au Canada, en Europe et aux États-Unis ont récemment proposé un nouveau concept en logistique, celui de l'Internet Physique (abbrégé PI ou π). Il s'agit d'une vision ambitieuse visant à transformer la façon dont les objets physiques sont transportés, stockés, fournis et utilisés – en s'inspirant de la façon dont les ordinateurs sont interconnectés via Internet. L'objectif est de réorganiser entièrement la distribution de marchandises d'une façon efficace, mais aussi soutenable des points de vue économique, environnemental et social. Les marchandises, encapsulées dans des π-conteneurs standardisés, seraient manutentionnées par des engins dédiés – π-déplaceurs, π-convoyeurs. Les π-crossdocks ou centres de transit, potentiellement multimodaux, faciliteraient ainsi le transbordement de camion à camion de façon automatisée et très rapide. Bien sûr le modèle développé au chapitre 5 pour les employés d'une plateforme logistique ne s'applique pas dans ce cas : grâce à la standardisation poussée des π-conteneurs, toutes les opérations de transbordement pourraient être automatisées à 100%. Cependant, les modèles de planification des camions dans les plateformes de cross-docking seraient tout à fait indispensables pour exploiter de telles installations de manière fluide.

physicalinternet initiative.org

APPENDIX

A

INTERVIEW GRID FOR PLATFORM VISITS

The interview grid used in every platform visit we made is displayed below. The grid is entirely in French since all the companies visited are French companies.

Guide d'entretien

Caractérisation d'un crossdock, identification de ses contraintes et objectifs

Entreprise : *Lieu :*

Caractérisation du Xdock

Secteur d'activité
Client(s)

Données de la plateforme
I L U T H E …
Surface
Nombre de portes
Nombre d'employés CDI/CDD
Nombre d'intérimaires

Amplitude d'ouverture
Horaires des équipes

Certifications

Flux
Qualification des flux: tendus stockés
Quantification des flux : gros moyens faibles
Volumes annuels
Volumes quotidiens
Variabilité

Mode de service
Porte exclusivement dédiées à l'entrée ou à la sortie
Gestion mixte
Portes exclusivement dédiées à une destination
Restrictions pour assigner les camions aux portes ?

Transport interne
Manuel quelles contraintes engins quelles contraintes RH – CACES

Automatique
Mixte

Système d'information
Quel système ?
Développement : externe interne
Support : externe interne

Contraintes

Temps de déplacement d'une porte à l'autre
Négligeables
Fonction de : distance congestion RH disponibles engins disponibles

Planification des ressources
Humaines
Tâches, métiers
Polyvalence ?
Modulation ?
Matérielles

Arrivées des camions
Tous en même temps/concentration
Répartition régulière

Heures estimées	Heures exactes	Tranche horaire

Gestion des aléas, des retards ?

Déchargement

Composition	mono-produits		multi-produits
Tps de déchargement	dépend du camion	± constant	entre 2 bornes
Départ des camions	non contraint	borne sup	entre 2 bornes

Interchangeabilité des produits
Produits spécifiquement assignés à des destinations
On a uniquement un nb et type de produit à charger par destination

Opérations à valeur ajoutée
Oui Non
Systématiques ? Temps

Stockage intermédiaire
Totalement interdit
Après déchargement
Au sol sur les quais — En zone de rétention — Sur racks
Avant rechargement
Au sol sur les quais — En zone de rétention — Sur racks
Capacités, coûts

Préemption
Interruption du chargement autorisé ? oui non
Coût du changement de camion

Heures de départ des camions sortants
Non contraint
Entre deux bornes Borne sup uniquement
Dès que camion chargé A heure fixe même si pas plein à heure fixe, doit être plein

Objectifs

Evaluation de performance

Somme pondérée des temps de prise en charge	Makespan
Retard maximum	Somme pondérée des retards
Total stocké	Niveau de stock max

Coups de fourche
Mouvements de camion
Engorgement

Besoins
Points difficiles, délicats, bloquants dans la planif ?

Objectifs de performance

B

DOCUMENTATION OF THE INSTANCE GENERATOR

Three files can be found at www.g-scop.fr/~gaujalg/XDockInstances2:

`instanceinput.txt`, a text file that enumerates the input data needed to generate an instance

`instancegenerator.jar` generates a new instance from the input data file placed in the same directory. The instance is visualized through an applet and saved in a file named `instanceData.txt`.

`instances.zip`, ZIP archive file containing all the instances used in this document (see Table 2.2), described with the same syntax as `instanceData.txt`.

B.1 INSTANCE GENERATOR: QUICK START

To generate an instance, follow the instructions:

1. Download `instanceInput.txt` and `instanceGenerator.jar` and place them in the same directory.
2. Ensure that Java is installed on your system.
3. Open `instanceInput.txt` with any text editor, modify it depending on your needs and save. The file enumerates, in this order:

   $|\mathcal{I}|$ number of inbound trucks to be scheduled;

   $|\mathcal{O}|$ number of outbound trucks to be scheduled;

   $|\mathcal{C}|$ number of clients to be served – pay attention to the fact that there should be enough trucks to serve the clients, so $|\mathcal{I}|$ must be greater than or equal to $|\mathcal{C}|$;

   $|\mathcal{H}|$ number of time units in the planning horizon;

   N^I number of inbound doors in the platform;

   N^O number of outbound doors in the platform;

   F outbound truck capacity, *i.e.* number of pallets that can be loaded in each outbound truck;

   M internal platform capacity, *i.e.* maximum number of pallets that can be transferred at each time unit.
4. Click on `instanceGenerator.jar`. An instance is generated from the input data. A window opens to show a visual representation of the instance; the description of the instance data is saved in the same directory, in a file named `instanceData.txt`.

B.2 INSTANCE GENERATOR INPUT

Figure B.1 gives an example of input file `instanceInput.txt` used to generate instance `17_1`. The numbers should be integers, separated by a single string of characters (no spaces are allowed in the comment lines). The order in which the different data elements are declared must be respected.

```
//|I|_nb_inbound_trucks
5
//|O|_nb_outbound_trucks
5
//|C|_nb_clients
3
//|H|_nb_hours
10
//NI_nb_inbound_doors
3
//NO_nb_outbound_doors
3
//F_outbound_trucks_capacity
33
//M_internal_platform_capacity
17
```

Figure B.1: Input file used to generate instance `17_1`

B.3 INSTANCE GENERATOR OUTPUT

B.3.1 *Description of the output file syntax and the embedded algorithms*

Figure B.2 gives an example of output file that is generated by the instance generator. The file is named `instanceData.txt` and placed in the same directory as the generator and the input file.

The first eight lines of the text document repeats the input data, in the same order as in `instanceInput.txt`. Next, we give details regarding the generated data:

THE INBOUND TRUCK MATRIX lists all the information about the inbound trucks. Each row represents a truck. The first column gives the earliest possible arrival time for the truck – this generator always uses 0 for this value. The second column gives the latest possible departure time – always $|\mathcal{H}|$ in the case of this generator. The third column gives the wished arrival time for this truck, noted as A, obtained by picking a random integer within the range $[0, |\mathcal{H}| - 1[$ (the truck cannot arrive at the last time unit or it could not be unloaded on time). The length of the wished presence time window is an integer noted L, randomly picked within the range $[1, |\mathcal{H}| - A[$ so that the

```
5
5
3
10
3
3
33
17

InboundTrucks
 0  10  1   8  1
 0  10  8  10  1
 0  10  5   6  1
 0  10  6  10  1
 0  10  8   9  1

OutboundTrucks
 0  10  7   8  3
 0  10  3  10  3
 0  10  9  10  3
 0  10  1   8  3
 0  10  3   5  3

Q_ic
  7   1  25
  8   6  19
  4  14  15
 10   9  14
  4   3  26

Z_co
 1  0  0  0  0
 0  1  0  0  0
 0  0  1  1  1
```

Figure B.2: Example of output file `instanceData.txt`

time window is at least one time unit long. The fourth column is the wished departure time, *i.e.* $A + L$. Finally, the last column is needed to generate matrix W^I: it is the minimum length of the presence slots enumerated in W^I. The generator calculates this as the minimum amount of time needed to unload the truck, *i.e.* its total number of pallets divided by M.

THE OUTBOUND TRUCK MATRIX groups exactly the same information as detailed above, but for the outbound trucks.

Q_{ic} MATRIX describes the contents of the incoming trucks. Each row represents an inbound truck, and each column corresponds to a client. A cell (i, c) in the matrix gives the number of pallets in inbound truck i in destination to client c. For the data to be consistent, the inbound quantity for each destination should be equivalent to the capacity of the outbound trucks for this destination. Matrix Q_{ic} is therefore generated with the algorithm detailed in algorithm B.1.

```
Fill the matrix with 0
foreach c ∈ C do
    qty = F ∑_{o∈O} Z_co    // Total number of pallets for client c
    for pallet from 0 to qty do
        repeat
            i = random integer within the range[0, |I|[
        until truckload(i) < F|O| / |I|    // A truck not full is found

        Q_ic = Q_ic + 1    // Add 1 pallet in truck i for client c

        Update truckload(i)
    end
end
```

Algorithm B.1: Generation of matrix Q_{ic}

MATRIX Z_{do} links the outbound trucks to the clients. To ensure that each client is served by at least one truck, the first $|\mathcal{C}|$ columns of the matrix are filled with "1" along the diagonal. The remaining columns are filled picking a random destination number for each truck left; algorithm B.2 details the procedure used.

```
Fill the matrix with 0
for o from 0 to |C| − 1 do
    Z_oo = 1
end
for o from |C| to |O| do
    c = random integer within the range [0, |C|[
    Z_co = 1
end
```

Algorithm B.2: Generation of matrix Z_{co}

B.3.2 *Example of visual representation*

Figure B.3 on the facing page gives an example of visual representation proposed by the instance generator, along with some explanations on what the different elements represent.

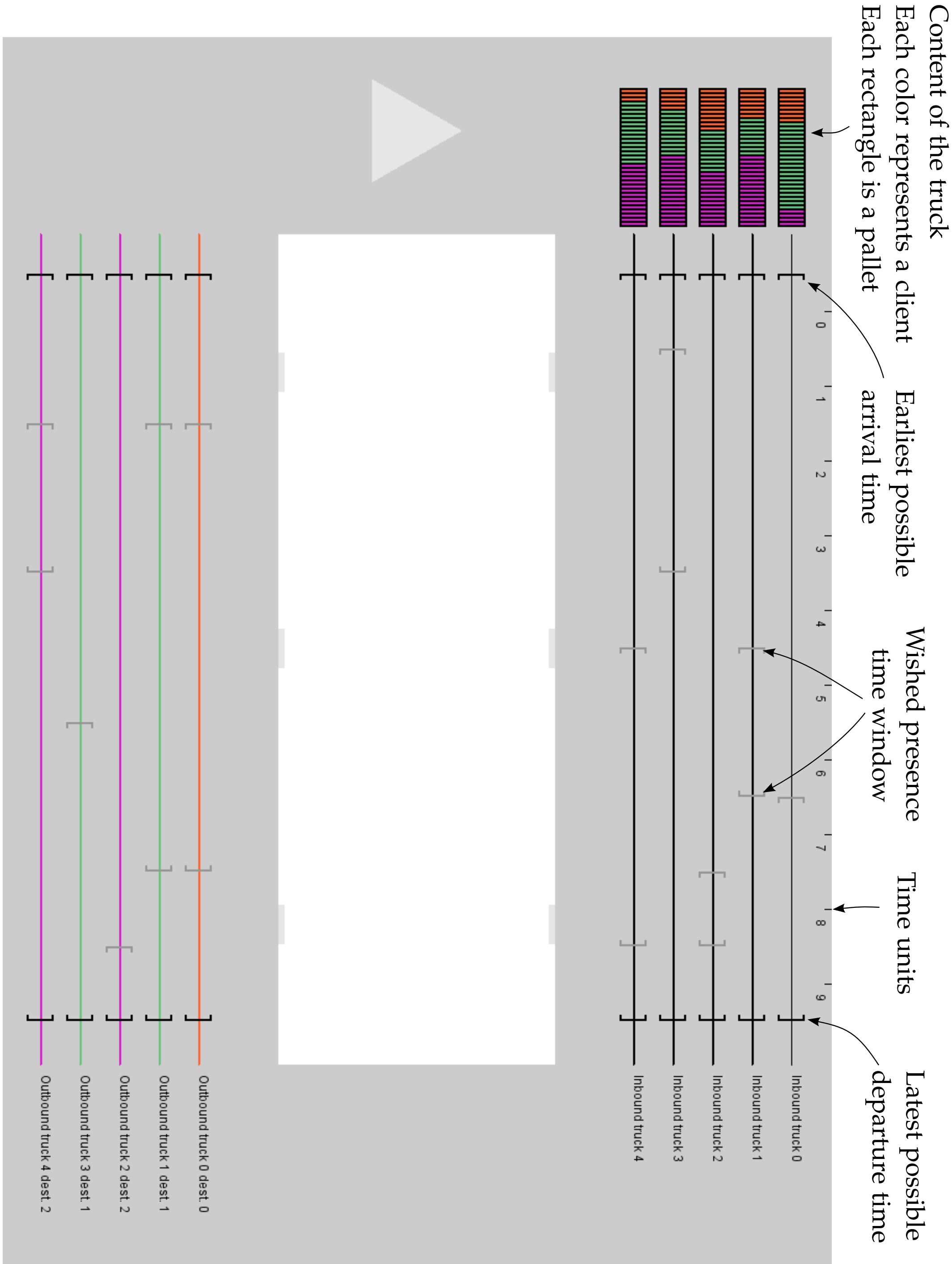


Figure B.3: Exemple of visual representation given by the instance generator

C INTEGER PROGRAMMING MODELS

Below are the detailed formulations of the integer programming models which do no appear explicitly in the main body of the dissertation.

$$
\begin{array}{lllr}
\min & \alpha_0\Pi^\alpha + \beta_0\Pi^\beta + \gamma_0\Pi^\gamma & & \\
& & & \\
\text{s.t.} & \Pi^\alpha = \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} P^I_{ik}w^I_{ik} & & (1') \\
& \Pi^\beta = \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} P^O_{ok}w^O_{ok} & & (2') \\
& \Pi^\gamma = \sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic} & & (3') \\
& \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} W^O_{okh}w^O_{ok} \leq N^O & \forall h\in\mathcal{H} & (5') \\
& x_{hio} + s^I_{hic} \leq F\sum_{k\in\mathcal{K}_i} W^I_{ikh}w^I_{ik} & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (6') \\
& x_{hio} + s^O_{ho} \leq F\sum_{k\in\mathcal{K}_o} W^O_{okh}w^O_{ok} & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (7') \\
& \sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co}x_{hio} + \sum_{h\in\mathcal{H}} s^I_{hic} = Q_{ic} & \forall i\in\mathcal{I}, c\in\mathcal{C} & (8') \\
& \sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s^O_{ho} = F & \forall o\in\mathcal{O} & (9') \\
& \sum_{o\in O} x_{hio} + \sum_{c\in\mathcal{C}} s^I_{hic} \leq M & \forall i\in\mathcal{I}, h\in\mathcal{H} & (10') \\
& \sum_{k\in\mathcal{K}_i} w^I_{ik} = 1 & \forall i\in\mathcal{I} & (11') \\
& s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s^I_{hic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{ho} & \forall c\in\mathcal{C}, h\in\mathcal{H}\setminus\{0\} & (13') \\
& s_{0c} = \sum_{i\in\mathcal{I}} s^I_{0ic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{0o} & \forall c\in\mathcal{C} & (14') \\
& & & \\
& x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc} \in \mathbb{N}^+ & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}, c\in\mathcal{C} & \\
& w^O_{ok} \in \{0,1\} & \forall o\in\mathcal{O}, k\in K &
\end{array}
$$

IP*1

$$
\begin{array}{lllr}
\min & \alpha_0\Pi^{\alpha} + \beta_0\Pi^{\beta} + \gamma_0\Pi^{\gamma} & & \\
\\
\text{s.t.} & \Pi^{\alpha} = \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} P^I_{ik}w^I_{ik} & & (1'') \\
 & \Pi^{\beta} = \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} P^O_{ok}w^O_{ok} & & (2'') \\
 & \Pi^{\gamma} = \sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic} & & (3'') \\
 & \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} W^I_{ikh}w^I_{ik} \leq N^I & \forall h\in\mathcal{H} & (4'') \\
 & x_{hio} + s^I_{hic} \leq F\sum_{k\in\mathcal{K}_i} W^I_{ikh}w^I_{ik} & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (6'') \\
 & x_{hio} + s^O_{ho} \leq F\sum_{k\in\mathcal{K}_o} W^O_{okh}w^O_{ok} & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (7'') \\
 & \sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co}x_{hio} + \sum_{h\in\mathcal{H}} s^I_{hic} = Q_{ic} & \forall i\in\mathcal{I}, c\in\mathcal{C} & (8'') \\
 & \sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s^O_{ho} = F & \forall o\in\mathcal{O} & (9'') \\
 & \sum_{o\in O} x_{hio} + \sum_{c\in\mathcal{C}} s^I_{hic} \leq M & \forall i\in\mathcal{I}, h\in\mathcal{H} & (10'') \\
 & \sum_{k\in\mathcal{K}_i} w^I_{ik} = 1 & \forall i\in\mathcal{I} & (11'') \\
 & s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s^I_{hic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{ho} & \forall c\in\mathcal{C}, h\in\mathcal{H}\setminus\{0\} & (13'') \\
 & s_{0c} = \sum_{i\in\mathcal{I}} s^I_{0ic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{0o} & \forall c\in\mathcal{C} & (14'') \\
\\
 & x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc} \in \mathbb{N}^+ & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}, c\in\mathcal{C} & \\
 & w^I_{ik} \in \{0,1\} & \forall i\in\mathcal{I}, k\in K &
\end{array}
$$

IP^{*2}

$$
\begin{array}{lllr}
\min & \Pi^{\gamma} = \sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic} & & \\
\\
\text{s.t.} & x_{hio} + s^I_{hic} \leq F\sum_{k\in\mathcal{K}_i} W^I_{ikh}w^I_{ik} & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (6''') \\
 & x_{hio} + s^O_{ho} \leq F\sum_{k\in\mathcal{K}_o} W^O_{okh}w^O_{ok} & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O} & (7''') \\
 & \sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co}x_{hio} + \sum_{h\in\mathcal{H}} s^I_{hic} = Q_{ic} & \forall i\in\mathcal{I}, c\in\mathcal{C} & (8''') \\
 & \sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s^O_{ho} = F & \forall o\in\mathcal{O} & (9''') \\
 & \sum_{o\in O} x_{hio} + \sum_{d\in D} s^I_{hic} \leq M & \forall i\in\mathcal{I}, h\in\mathcal{H} & (10''') \\
 & s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s^I_{hic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{ho} & \forall c\in\mathcal{C}, h\in\mathcal{H}\setminus\{0\} & (13''') \\
 & s_{0c} = \sum_{i\in\mathcal{I}} s^I_{0ic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{0o} & \forall c\in\mathcal{C} & (14''') \\
\\
 & x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc} \in \mathbb{N}^+ & \forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}, c\in\mathcal{C} &
\end{array}
$$

IP^{*3}

(IP1)$^{\text{D1}}$ is a version of IP1 (on page 55) with $\min r^I$ as objective function, and constraint (15)$^{\text{D1}}$ added, as well as constraint (1)$^{\text{D1}}$ to ensure that the search is restricted to the optimal solutions of IP1.

min	r^I		
s.t.	$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} p^I_{ik}w^I_{ik} \le \Pi^\alpha_0$		$(1)^{D1}$
	$x_{hio} + s^I_{hic} \le F\sum_{k\in\mathcal{K}_i} W^I_{ikh}w^I_{ik}$	$\forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}$	$(6''')$
	$x_{hio} + s^O_{ho} \le F\sum_{k\in\mathcal{K}_o} W^O_{okh}w^O_{ok}$	$\forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}$	$(7''')$
	$\sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co}x_{hio} + \sum_{h\in\mathcal{H}} s^I_{hic} = Q_{ic}$	$\forall i\in\mathcal{I}, c\in\mathcal{C}$	$(8''')$
	$\sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s^O_{ho} = F$	$\forall o\in\mathcal{O}$	$(9''')$
	$\sum_{o\in O} x_{hio} + \sum_{d\in D} s^I_{hic} \le M$	$\forall i\in\mathcal{I}, h\in\mathcal{H}$	$(10''')$
	$s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s^I_{hic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{ho}$	$\forall c\in\mathcal{C}, h\in\mathcal{H}\setminus\{0\}$	$(13''')$
	$s_{0c} = \sum_{i\in\mathcal{I}} s^I_{0ic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{0o}$	$\forall c\in\mathcal{C}$	$(14''')$
	$\sum_{h\in\mathcal{H},i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} W^I_{ikh}w^I_{ik} \le r^I N^I\vert\mathcal{H}\vert$		$(15)^{D1}$
	$x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc} \in \mathbb{N}^+$	$\forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}, c\in\mathcal{C}$	

$(\text{IP1})^{D1}$

$(\text{IP}^{*1})^{D1}$ is a version of IP* (on page 46) with $\min r^O$ as objective function, and constraint $(16)^{D1}$ added, as well as constraints $(2)^{D1}$ and $(3)^{D1}$ to restrict the search to the optimal solutions of IP^{*1}.

min	r^O		
s.t.	$\sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} p^O_{ok}w^O_{ok} \le \Pi^\beta_0$		$(2)^{D1}$
	$\sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic} \le \Pi^\gamma_0$		$(3)^{D1}$
	$\Pi^\alpha_0 = \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} P^I_{ik}w^I_{ik}$		(1)
	$\Pi^\beta_0 = \sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} P^O_{ok}w^O_{ok}$		(2)
	$\Pi^\gamma_0 = \sum_{h\in\mathcal{H},i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic}$		(3)
	$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} W^I_{ikh}w^I_{ik} \le N^I$	$\forall h\in\mathcal{H}$	(4)
	$\sum_{o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} W^O_{okh}w^O_{ok} \le N^O$	$\forall h\in\mathcal{H}$	(5)
	$x_{hio} + s^I_{hic} \le F\sum_{k\in\mathcal{K}_i} W^I_{ikh}w^I_{ik}$	$\forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}$	(6)
	$x_{hio} + s^O_{ho} \le F\sum_{k\in\mathcal{K}_o} W^O_{okh}w^O_{ok}$	$\forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}$	(7)
	$\sum_{h\in\mathcal{H},o\in\mathcal{O}} Z_{co}x_{hio} + \sum_{h\in\mathcal{H}} s^I_{hic} = Q_{ic}$	$\forall i\in\mathcal{I}, c\in\mathcal{C}$	(8)
	$\sum_{i\in\mathcal{I},h\in\mathcal{H}} x_{hio} + \sum_{h\in\mathcal{H}} s^O_{ho} = F$	$\forall o\in\mathcal{O}$	(9)
	$\sum_{o\in O} x_{hio} + \sum_{c\in\mathcal{C}} s^I_{hid} \le M$	$\forall i\in\mathcal{I}, h\in\mathcal{H}$	(10)
	$\sum_{k\in\mathcal{K}_i} w^I_{ik} = 1$	$\forall i\in\mathcal{I}$	(11)
	$\sum_{k\in\mathcal{K}_o} w^O_{ok} = 1$	$\forall o\in\mathcal{O}$	(12)
	$s_{hc} = s_{(h-1)c} + \sum_{i\in\mathcal{I}} s^I_{hic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{ho}$	$\forall c\in\mathcal{C}, h\in\mathcal{H}\setminus\{0\}$	(13)
	$s_{0c} = \sum_{i\in\mathcal{I}} s^I_{0ic} - \sum_{o\in\mathcal{O}} Z_{co}s^O_{0o}$	$\forall c\in\mathcal{C}$	(14)
	$\sum_{h\in\mathcal{H},o\in\mathcal{O}}\sum_{k\in\mathcal{K}_o} W^O_{okh}w^O_{ok} \le r^O N^O\vert\mathcal{H}\vert$		$(16)^{D1}$
	$x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc} \in \mathbb{N}^+$	$\forall h\in\mathcal{H}, i\in\mathcal{I}, o\in\mathcal{O}, c\in\mathcal{C}$	
	$w^I_{ik}, w^O_{ok} \in \{0,1\}$	$\forall i\in\mathcal{I}, o\in\mathcal{O}, k\in K$	

$(\text{IP}^{*1})^{D1}$

D STANDARD TIME CALCULATIONS

This appendix details how the standard time used for simulation are obtained.

D.1 DISTANCES

The distances are not explicitly taken into account in our model, but they should appear implicitly in the processing times. To calculate the processing time, the following assumptions are made regarding distances:

PLATFORM WIDTH. According to Bartholdi and Gue [19], I-shaped platforms are about 60 to 120 feet wide. In the case of small platforms, the width of the platform is therefore 60 feet, *i. e.* 18 meters.

PLATFORM LENGTH. Bartholdi and Gue also state that the space between doors is generally 12 feet. The length of a platform with 3 inbound and 3 outbound doors will therefore be $12 \times 3 = 36$ feet, *i. e.* 11 meters.

TRUCK SIZE. We assume that a standard truck capable of carrying 33 European pallets of 800×1200 mm, measures 8.8×3.6 meters.

DOOR-DOCK DISTANCE. We assume that there is 1 meter between the truck door and the dock.

In the best case, a pallet has to go from a given inbound dock to the one just opposite: the distance from dock to dock is then 18 meters. In the worse case, the pallet has to go from a side dock to the dock on the opposite side in diagonal: the distance is then $\sqrt{60^2 + 36^2} = 70$ feet, *i. e.* about 21 meters. The distances to be crossed during transfer and unloading are presented in Figure D.1.

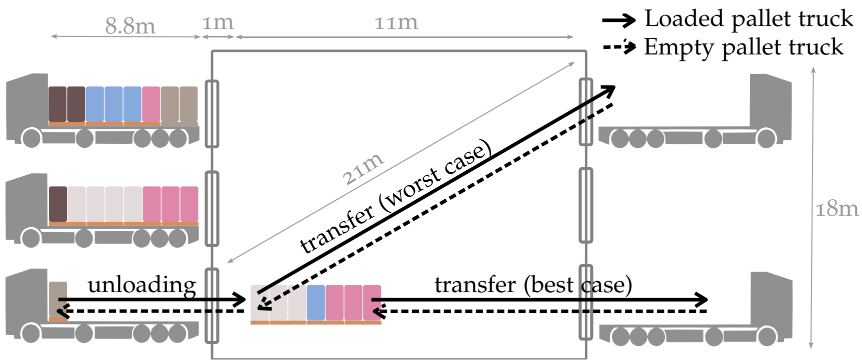


Figure D.1: Distances for unloading and transfer

D.2 STANDARD TIMES

The standard times in Table D.1 and D.2 are taken from Gauvreau [78].

Operation	In a clear area (s)	In a congested area (s)
Load a pallet	70	88
Start the loaded pallet truck	8	8
Stop the loaded pallet truck	8	8
Unload the pallet truck	34	34
Total set-up time	120	138

Source: Gauvreau [78]

Table D.1: Standard times: set-up

Operation	Load $<$ 60kg	Load $>$ 60kg	Empty
Move a pallet truck	2.36 s/m	3.34 s/m	2.36 s/m

Source: Gauvreau [78]

Table D.2: Standard times: moving a pallet truck

Operation	Worse conditions	Best conditions
Scan a pallet	40 seconds	90 seconds

Source: measures taken in an industrial context

Table D.3: Standard times: scanning a pallet

D.3 DETAILS OF THE CALCULATIONS

The distance and standard time enable to calculate the following values:

UNLOADING IN THE BEST CASE. In the best case, the pallet truck travels a distance of 9.80 m at a speed of 2.36 s/m when loaded, then the same distance empty to come back to the next pallet. Adding the set-up time and the time needed to scan the pallet, the total time needed to unload a pallet in the best case is:

$$9.80 \times 2.36 + 9.80 \times 2.36 + 120 + 40 = 202.26 \text{ s} = 3.5 \text{ min}$$

UNLOADING IN THE WORST CASE. In the worst case, the pallet truck travels a distance of 9.80 m at a speed of 3.34 s/m when loaded with a heavy pallet, then the same distance empty. Adding the set-up and scanning time, the total time needed to unload a pallet in the worst case is:

$$9.80 \times 3.34 + 9.80 \times 2.36 + 138 + 90 = 283.86 \text{ s} = 4.7 \text{ min}$$

TRANSFER IN THE BEST CASE. In the best case, the pallet truck travels a distance of 11 m at a speed of 2.36 s/m when loaded, then the same distance empty to come back to the next pallet. Adding the set-up time, the total time needed to transfer a pallet in the best case is:

$$11 \times 2.36 + 11 \times 2.36 + 120 = 171.92 \text{ seconds} = 2.8 \text{ minutes}$$

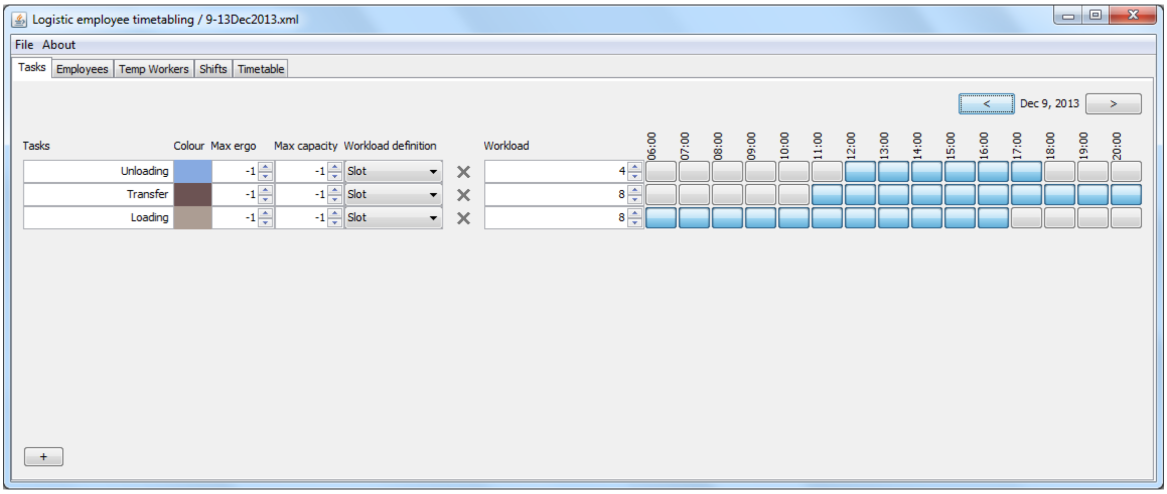
TRANSFER IN THE WORST CASE. In the best case, the pallet truck travels a distance of 21 m at a speed of 3.34 s/m when loaded, then the same distance empty to come back to the next pallet. Adding the set-up time, the total time needed to transfer a pallet in the worst case is:

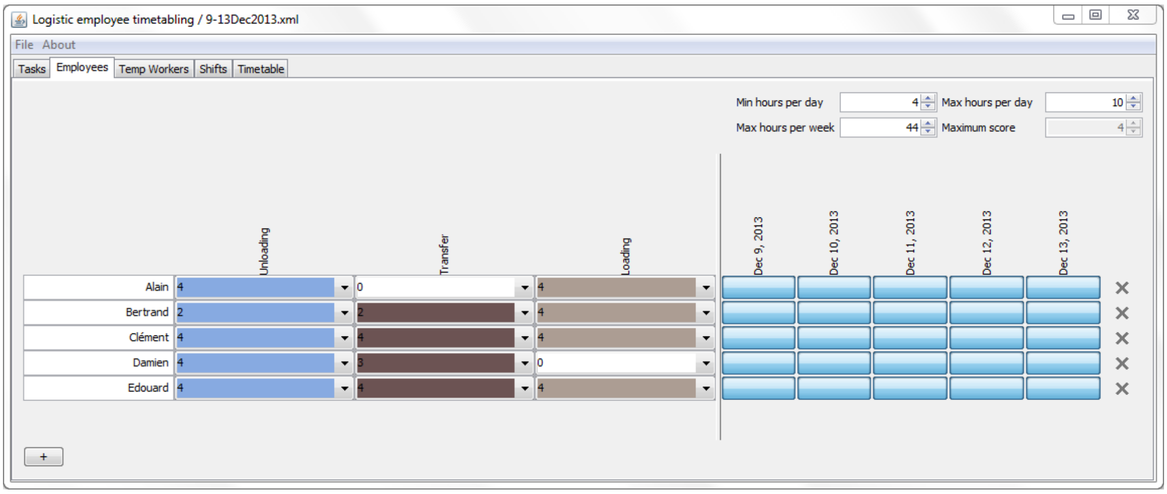
$$21 \times 3.34 + 21 \times 2.36 + 138 = 257.70 \text{ seconds} = 4.3 \text{ minutes}$$

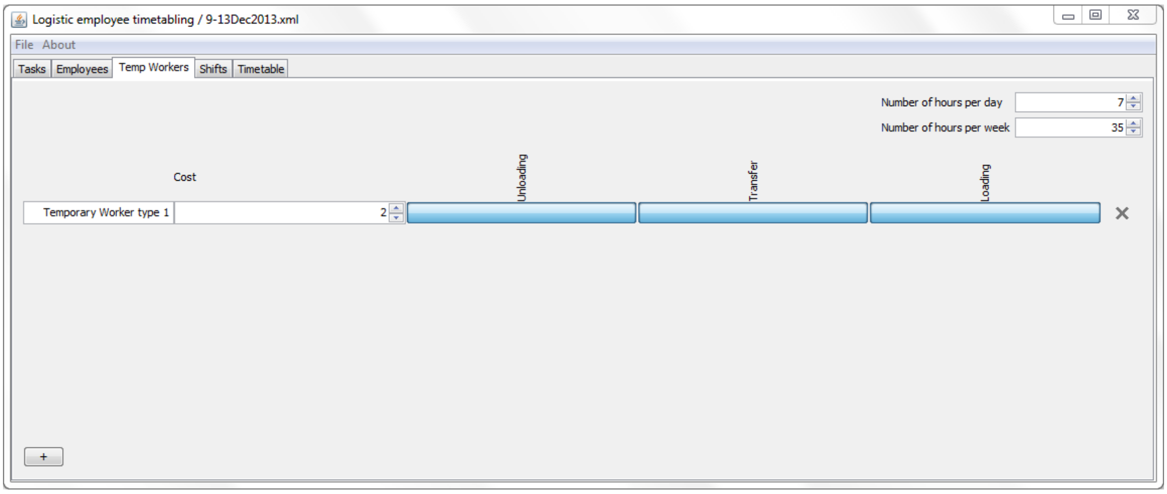
The values obtained are the values of parameters a and b in the triangular distribution in Table 3.1.

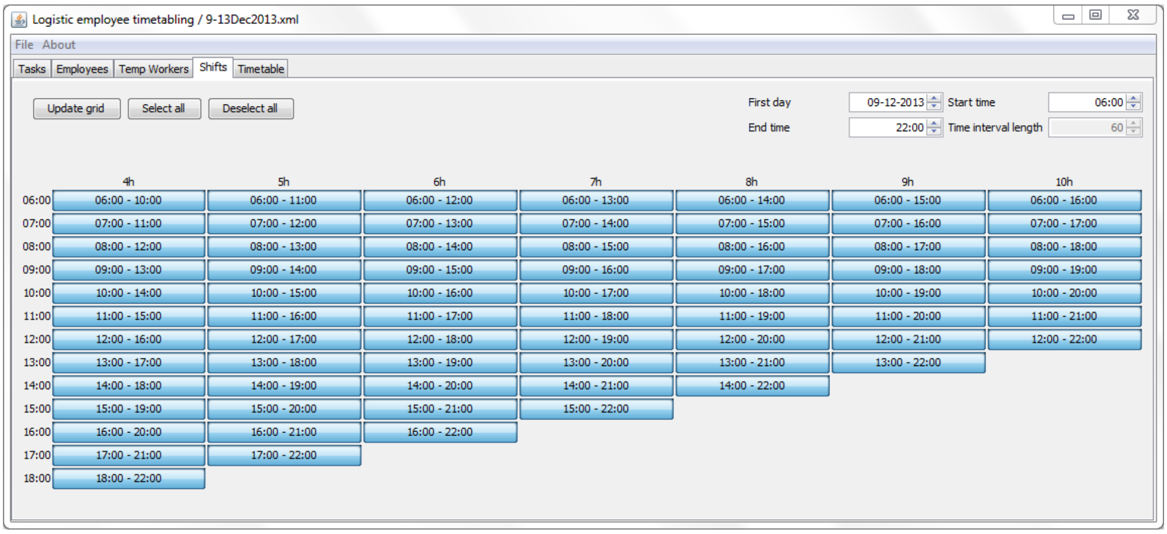
E

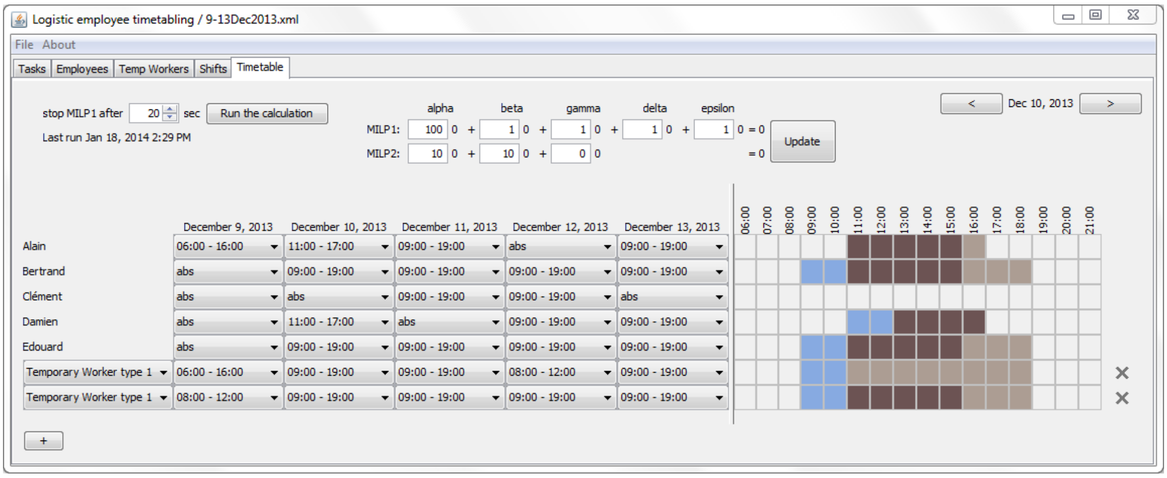
GRAPHICAL INTERFACE FOR THE TIMETABLING TOOL








Logistic employee timetabling / 9-13Dec2013.xml
File About
Tasks Employees Temp Workers Shifts Timetable
Update grid
Select all
Deselect all
First day 09-12-2013
Start time 06:00
End time 22:00
Time interval length 60
4h 5h 6h 7h 8h 9h 10h
06:00 06:00 - 10:00 06:00 - 11:00 06:00 - 12:00 06:00 - 13:00 06:00 - 14:00 06:00 - 15:00 06:00 - 16:00
07:00 07:00 - 11:00 07:00 - 12:00 07:00 - 13:00 07:00 - 14:00 07:00 - 15:00 07:00 - 16:00 07:00 - 17:00
08:00 08:00 - 12:00 08:00 - 13:00 08:00 - 14:00 08:00 - 15:00 08:00 - 16:00 08:00 - 17:00 08:00 - 18:00
09:00 09:00 - 13:00 09:00 - 14:00 09:00 - 15:00 09:00 - 16:00 09:00 - 17:00 09:00 - 18:00 09:00 - 19:00
10:00 10:00 - 14:00 10:00 - 15:00 10:00 - 16:00 10:00 - 17:00 10:00 - 18:00 10:00 - 19:00 10:00 - 20:00
11:00 11:00 - 15:00 11:00 - 16:00 11:00 - 17:00 11:00 - 18:00 11:00 - 19:00 11:00 - 20:00 11:00 - 21:00
12:00 12:00 - 16:00 12:00 - 17:00 12:00 - 18:00 12:00 - 19:00 12:00 - 20:00 12:00 - 21:00 12:00 - 22:00
13:00 13:00 - 17:00 13:00 - 18:00 13:00 - 19:00 13:00 - 20:00 13:00 - 21:00 13:00 - 22:00
14:00 14:00 - 18:00 14:00 - 19:00 14:00 - 20:00 14:00 - 21:00 14:00 - 22:00
15:00 15:00 - 19:00 15:00 - 20:00 15:00 - 21:00 15:00 - 22:00
16:00 16:00 - 20:00 16:00 - 21:00 16:00 - 22:00
17:00 17:00 - 21:00 17:00 - 22:00
18:00 18:00 - 22:00


Logistic employee timetabling / 9-13Dec2013.xml
File About
Tasks Employees Temp Workers Shifts Timetable
stop MILP1 after 20 sec
Run the calculation
Last run Jan 18, 2014 2:29 PM
alpha beta gamma delta epsilon
MILP1: 100 0 + 1 0 + 1 0 + 1 0 + 1 0 = 0
MILP2: 10 0 + 10 0 + 0 0 = 0
Update
< Dec 10, 2013 >
December 9, 2013 December 10, 2013 December 11, 2013 December 12, 2013 December 13, 2013
Alain 06:00 - 16:00 11:00 - 17:00 09:00 - 19:00 abs 09:00 - 19:00
Bertrand abs 09:00 - 19:00 09:00 - 19:00 09:00 - 19:00 09:00 - 19:00
Clément abs abs 09:00 - 19:00 09:00 - 19:00 abs
Damien abs 11:00 - 17:00 abs 09:00 - 19:00 09:00 - 19:00
Edouard abs 09:00 - 19:00 09:00 - 19:00 09:00 - 19:00 09:00 - 19:00
Temporary Worker type 1 06:00 - 16:00 09:00 - 19:00 09:00 - 19:00 08:00 - 12:00 09:00 - 19:00
Temporary Worker type 1 08:00 - 12:00 09:00 - 19:00 09:00 - 19:00 09:00 - 19:00 09:00 - 19:00
06:00 07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00
+

BIBLIOGRAPHY

[1] Acar, M. K. 2004. Robust dock assignments at less-than-truckload terminals. M.S. thesis, University of South Florida.

[2] Acar, M. K., Yalcin, A., and Yankov, D. 2012. Robust door assignment in less-than-truckload terminals. *Computers & Industrial Engineering 63,* 4, 729–738.

[3] Agustina, D., Lee, C. K. M., and Piplani, R. 2012. Scheduling and vehicle routing model of cross docking. In *International Conference on Future Information Technology and Management Science & Engineering*. Hong Kong, 373–378.

[4] Agustina, D., Lee, C. K. M., and Piplani, R. 2014. Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics 152*, 29–41.

[5] Ahuja, R. K., Magnati, T. L., and Orlin, J. B. 1993. *Network flows: Theory, algorithms and applications*. Prentice Hall, Upper Saddle River, New Jersey.

[6] Aickelin, U. and Adewunmi, A. 2006. Simulation optimization of the crossdock door assignment problem. Tech. rep., University of Nottingham, Leamington Spa, UK.

[7] Almeder, C., Preusser, M., and Hartl, R. F. 2009. Simulation and optimization of supply chains: alternative or complementary approaches? *OR Spectrum 31*, 95–119.

[8] Alpan, G., Bauchau, S., Larbi, R., and Penz, B. 2008. Optimal operations scheduling in a crossdock with multi strip and multi stack doors. In *International Conference on Computers and Industrial Engineering*. Beijing, China, 1168–1176.

[9] Alpan, G., Ladier, A.-L., Larbi, R., and Penz, B. 2011. Heuristic solutions for transshipment problems in a multiple door cross docking warehouse. *Computers & Industrial Engineering 61,* 2, 402–408.

[10] Alpan, G., Larbi, R., and Penz, B. 2011. A bounded dynamic programming approach to schedule operations in a cross docking platform. *Computers & Industrial Engineering 60,* 3, 385–396.

[11] Álvarez Pérez, G. A., González-Velarde, J. L., and Fowler, J. W. 2009. Crossdocking – just in time scheduling: an alternative solution approach. *Journal of the Operational Research Society 60,* 4, 554–564.

[12] Apte, U. M. and Viswanathan, S. 2000. Effective cross docking for improving distribution efficiencies. *International Journal of Logistics Research and Applications 3,* 3, 291–302.

[13] Artigues, C., Gendreau, M., and Rousseau, L.-M. 2007. A flexible model and a hybrid exact method for integrated employee timetabling and production scheduling. *Lecture Notes in Computer Science 3867,* Practice and Theory of Automated Timetabling VI, 67–84.

[14] Artigues, C., Gendreau, M., Rousseau, L.-M., and Vergnaud, A. 2009. Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound. *Computers & Operations Research 36,* 8, 2330–2340.

[15] Baker, P. and Halim, Z. 2007. An exploration of warehouse automation implementations: cost, service and flexibility issues. *Supply Chain Management: An International Journal 12*, 129–138.

[16] Baptiste, P. and Maknoon, M. Y. 2007. Cross-docking: Scheduling of incoming and outgoing semi-trailers. In *International Conference on Production Research*. Valparaiso, Chile, 1–6.

[17] Bard, J. F., Binici, C., and DeSilva, A. H. 2003. Staff scheduling at the United States Postal Service. *Computers & Operations Research 30,* 5, 745–771.

[18] Bartholdi, J. J. and Gue, K. R. 2000. Reducing labor costs in an LTL crossdocking terminal. *Operations Research 48,* 6, 823–832.

[19] Bartholdi, J. J. and Gue, K. R. 2004. The best shape for a crossdock. *Transportation Science 38,* 2, 235–244.

[20] Bartz-Beielstein, T., Chmielewski, A., Janas, M., Naujoks, B., and Scheffermann, R. 2006. Optimizing door assignment in LTL-terminals by evolutionary multiobjective algorithms. In *IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation*. Vancouver BC, US, 348–354.

[21] Bellanger, A., Hanafi, S., and Wilbaut, C. 2012. Three-stage hybrid-flowshop model for cross docking. *Computers & Operations Research 40,* 4, 1109–1121.

[22] Berghman, L., Briand, C., Leus, R., and Lopez, P. 2012. The truck scheduling problem at cross-docking terminals. In *International Conference on Project Management and Scheduling*. Louvain, Belgium.

[23] Bermúdez, R. and Cole, M. H. 2001. A genetic algorithm approach to door assignments in breakbulk terminals. Tech. rep., Mack-Blackwell Rural Transportation Center, University of Arkansas.

[24] Boloori Arabani, A., Fatemi Ghomi, S. M. T., and Zandieh, M. 2010. A multi-criteria cross-docking scheduling with just-in-time approach. *The International Journal of Advanced Manufacturing Technology 49,* 5, 741–756.

[25] Boloori Arabani, A., Fatemi Ghomi, S. M. T., and Zandieh, M. 2011. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert Systems with Applications 38,* 3, 1964–1979.

[26] Boloori Arabani, A., Zandieh, M., and Fatemi Ghomi, S. M. T. 2011. Multi-objective genetic-based algorithms for a cross-docking scheduling problem. *Applied Soft Computing 11,* 8, 4954–4970.

[27] Boloori Arabani, A., Zandieh, M., and Fatemi Ghomi, S. M. T. 2012. A cross-docking scheduling problem with sub-population multi-objective algorithms. *The International Journal of Advanced Manufacturing Technology 58,* 5-8, 741–761.

[28] Boysen, N. 2010. Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research 37,* 1, 32–41.

[29] Boysen, N., Briskorn, D., and Tschöke, M. 2010. Truck scheduling in cross docking terminals with fixed outbound departures. Tech. Rep. 10/2010, Friedrich-Schiller-University Jena, School of Economics and Business Administration, Jena, Germany.

[30] Boysen, N., Briskorn, D., and Tschöke, M. 2012. Truck scheduling in cross-docking terminals with fixed outbound departures. *OR Spectrum 35,* 2, 479–504.

[31] Boysen, N. and Fliedner, M. 2010. Cross dock scheduling: Classification, literature review and research agenda. *Omega 38,* 6, 413–422.

[32] Boysen, N., Fliedner, M., Jaehn, F., and Pesch, E. 2012. A survey on container processing in railway yards. *Transportation Science 47,* 3, 312–329.

[33] Boysen, N., Fliedner, M., and Scholl, A. 2010. Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum 32,* 1, 135–161.

[34] Bozer, Y. A. and Carlo, H. J. 2008. Optimizing inbound and outbound door assignments in less-than-truckload crossdocks. *IIE Transactions 40,* 11, 1007–1018.

[35] Brown, A. M. 2003. Improving the efficiency of hub operations in a less-than-truckload distribution network. M.S. thesis, Virginia Polytechnic Institute and State University.

[36] Brucker, P., Qu, R., and Burke, E. K. 2011. Personnel scheduling: models and complexity. *European Journal of Operational Research 210,* 3, 467–473.

[37] Burke, E. K., De Causmaecker, P., Vanden Berghe, G., and Van Landeghem, H. 2004. The state of the art of nurse rostering. *Journal of Scheduling 7,* 6, 441–499.

[38] Campbell, G. M. and Diaby, M. 2002. Development and evaluation of an assignment heuristic for allocating cross-trained workers. *European Journal of Operational Research 138,* 1, 9–20.

[39] Canon, L.-C. and Jeannot, E. 2010. Evaluation and optimization of the robustness of DAG schedules in heterogeneous environments. *IEEE Transactions on Parallel and Distributed Systems 21,* 4, 532–546.

[40] Carlo, H. J. and Bozer, Y. A. 2011. Analysis of optimum shape and door assignment problems in rectangular unit-load crossdocks. *International Journal of Logistics Research and Applications 14,* 3, 149–163.

[41] Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. 2012. A survey on workforce scheduling and routing problems. In *International Conference on the Practice and Theory of Automated Timetabling*. Son, Norway, 283–302.

[42] Chen, F. and Lee, C.-Y. 2009. Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research 193,* 1, 59–72.

[43] Chen, F. and Song, K. 2009. Minimizing makespan in two-stage hybrid cross docking scheduling problem. *Computers & Operations Research 36,* 6, 2066–2073.

[44] Chen, R., Fan, B., and Tang, G. 2009. Scheduling problems in cross docking. In *Combinatorial Optimization and Applications*, D.-Z. Du, X. Hu, and P. Pardalos, Eds. Lecture Notes in Computer Science, vol. 5573. Springer Berlin Heidelberg, 421–429.

[45] Chiang, W.-Y. and Fox, M. S. 1990. Protection against uncertainty in a deterministic schedule. In *International Conference on Expert Systems in Production and Operations Management*. Hilton Head, South Carolina.

[46] Chmielewski, A., Naujoks, B., Janas, M., and Clausen, U. 2009. Optimizing the door assignment in LTL-terminals. *Transportation Science 43 (2)*, 198–210.

[47] Choy, K. L., Chow, H. K. H., Poon, T. C., and Ho, G. T. S. 2012. Cross-dock job assignment problem in space-constrained industrial logistics distribution hubs with a single docking zone. *International Journal of Production Research 50,* 9, 2439–2450.

[48] Clausen, U., Goedicke, I., Mest, L., and Wohlgemuth, S. 2012. Combining simulation and optimization to improve LTL traffic. *Procedia - Social and Behavioral Sciences 48,* 1, 1993–2002.

[49] Cohen, Y. and Keren, B. 2008. A simple heuristic for assigning doors to trailers in cross-docks. In *International Conference on Industrial Logistics*. Tel Aviv, Israel, 1–14.

[50] Cohen, Y. and Keren, B. 2009. Trailer to door assignment in a synchronous cross-dock operation. *International Journal of Logistics Systems and Management 5*, 574–590.

[51] Cook, R. L., Gibson, B., and MacCurdy, D. 2005. A lean approach to cross docking. *Supply Chain Management review*, 54–59.

[52] CSCMP. 2013. Glossary of Terms. Tech. rep., Council of Supply Chain Management Professionals.

[53] Dahmen, S. and Rekik, M. 2012. Solving multi-activity personalized shift scheduling problems with a hybrid heuristic. Tech. rep., Interuniversity Research Centre on Enterprise Networks, Logistics and transportation, Québec, Canada.

[54] Dandal, S. 2005. Staging approaches to reduce overall cost in a cross-dock environment. M.S. thesis, University of Missouri, Columbia.

[55] Davenport, A. J. and Beck, C. J. 2000. A survey of techniques for scheduling with uncertainty. Unpublished manuscript. Available from http://tidel.mie.utoronto.ca/publications.php.

[56] Davenport, A. J., Gefflot, C., and Beck, C. J. 2001. Slack-based techniques for robust schedules. In *International Conference on Principles and Practice of Constraint Programming*. Paphos, Cyprus.

[57] Davoudpour, H., Hooshangi-Tabrizi, P., and Hoseinpour, P. 2012. A genetic algorithm for truck scheduling in cross docking systems. *Journal of American Science 8,* 2, 96–99.

[58] De Bruecker, P., van den Bergh, J., Beliën, J., and Demeulemeester, E. 2014. Workforce planning incorporating skills : state of the art. Tech. rep., Faculty of economics and business, KU Leuven, Leuven, Belgium.

[59] De Causmaecker, P., Demeester, P., Vanden Berghe, G., and Verbeke, B. 2004. Analysis of real-world personnel scheduling problems. In *International Conference on the Practice and Theory of Automated Timetabling*. Pittsburgh, United States, 183–198.

[60] de Leeuw, S. and Wiers, V. C. 2014. Warehouse manpower planning strategies in times of financial crisis: evidence from logistics service providers and retailers in the Netherlands. *Production Planning & Control*, 1–10.

[61] Detienne, B., Péridy, L., Pinson, E., and Rivreau, D. 2009. Cut generation for an employee timetabling problem. *European Journal of Operational Research 197,* 3, 1178–1184.

[62] Deutsche Post DHL. 2012. *Delivering tomorrow – Logistics 2050, a Scenario Study*, 1st ed. Deutsche Post AG, Bonn, Germany.

[63] Dondo, R. and Cerdá, J. 2014. A monolithic approach to vehicle routing and operations scheduling of a cross-dock system with multiple dock doors. *Computers & Chemical Engineering 63*, 184–205.

[64] Dorndorf, U., Drexl, A., Nikulin, Y., and Pesch, E. 2007. Flight gate scheduling: State-of-the-art and recent developments. *Omega 35,* 3, 326–334.

[65] Eitzen, G., Panton, D., and Mills, G. 2004. Multi-skilled workforce optimisation. *Annals of Operations Research 127,* 1, 359–372.

[66] Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., and Sier, D. 2004. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research 127,* 1, 21–144.

[67] Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. 2004. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research 153,* 1, 3–27.

[68] Ertek, G. 2005. A tutorial on crossdocking. In *Proceedings of 3rd International Logistics & Supply Chain Congress*. Istanbul, Turkey.

[69] Ertek, G. 2012. Crossdocking insights from a third party logistics firm in Turkey. In *Managing supply chains on the Silk Road: Strategy, performance, and risk*, C. Haksöz, S. Seshadri, and A. V. Iyer, Eds.

[70] Faber, N., de Koster, R. M. B. M., and van de Velde Steef L. 2002. Linking warehouse complexity to warehouse planning and control structure: an exploratory study of the use of warehouse management information systems. *International Journal of Physical Distribution & Logistics Management 32*, 5, 381–395.

[71] Fazel Zarandi, M. H., Khorshidian, H., and Akbarpour Shirazi, M. 2014. A constraint programming model for the scheduling of JIT cross-docking systems with preemption. *Journal of Intelligent Manufacturing*, 1–17.

[72] Figueira, J., Greco, S., and Ehrgott, M., Eds. 2004. *Multiple criteria decision analysis: state of the art surveys*. Springer Science & Business Media.

[73] Forouharfard, S. and Zandieh, M. 2010. An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. *The International Journal of Advanced Manufacturing Technology 51*, 9, 1179–1193.

[74] Gabrel, V., Murat, C., and Thièle, A. 2013. Recent advances in robust optimization : An overview. Tech. rep., Université Paris-Dauphine, Lamsade, Paris.

[75] Gambardella, L. M., Rizzoli, A. E., and Zaffalon, M. 1998. Simulation and planning of an intermodal container terminal. *Simulation 71*, 2, 107–116.

[76] Gao, H. 1996. Building robust schedules using temporal protection – an empirical study of constraint based scheduling under machine failure uncertainty. Master thesis, University of Toronto, Canada.

[77] Garey, M. R. and Johnson, D. S. 1979. *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman.

[78] Gauvreau, P. 2009. *Les standards de temps logistiques, la méthode SMB*. Editions Celse.

[79] Ghobadian, E., Tavakkoli-Moghaddam, R., Javanshir, H., and Naderi-Beni, M. 2012. Scheduling trucks in cross docking systems with temporary storage and dock repeat truck holding pattern using grasp method. *International Journal of Industrial Engineering Computations*, 777–786.

[80] Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research 13*, 5, 533–549.

[81] Golias, M. M., Ivey, S., Haralambides, H. E., and Saharidis, G. K. D. 2010. Maximizing throughput and minimizing tardiness and earliness at a cross dock facility: Biobjective formulation. In *Transportation Research Board 89th Annual Meeting*. Washington DC, United States, 1–14.

[82] Golias, M. M., Ivey, S., Ji, K., and Lipinski, M. E. 2010. A bi-objective model to minimize service and storage time at a cross dock facility. In *51st Annual Transportation Research Forum*. Arlington, Virginia.

[83] Golias, M. M., Saharidis, G. K. D., Boile, M., and Theofanis, S. 2012. Scheduling of inbound trucks at a cross-docking facility: Bi-objective vs bi-level modeling approaches. *International Journal of Information Systems and Supply Chain Management 5,* 1, 20–37.

[84] Golias, M. M., Saharidis, G. K. D., Ivey, S., and Haralambides, H. E. 2013. Advances in truck scheduling at a cross dock facility. *International Journal of Information Systems and Supply Chain Management 6,* 3, 40–62.

[85] Golias, M. M., Saharidis, G. K. D., Ivey, S., and Lipinski, M. E. 2011. Risk averse truck scheduling at a cross-dock facility. In *Transportation Research Board 90th Annual Meeting*. Washington DC, USA, 1–12.

[86] Graham, D. D. 2003. Warehouse of the future. *Frontline Solutions 4*, 20–26.

[87] Granlund, A. 2011. Competitive internal logistics systems through automation. Ph.D. thesis, Mälardalen University.

[88] Gue, K. R. 1999. The effects of trailer scheduling on the layout of freight terminals. *Transportation Science 33,* 4, 419–428.

[89] Gue, K. R. 2007. Warehouses Without Inventory. *International Commerce Review 7,* 2, 124–132.

[90] Gue, K. R., Akcali, E., Erera, A., Ferrell, B., and Forger, G. 2014. Material handling & logistics US roadmap.

[91] Gue, K. R. and Kang, K. 2001. Staging queues in material handling and transportation systems. In *Proceedings of the Winter Simulation Conference*. Washington, DC, USA, 1104–1108.

[92] Guignard, M. and Hahn, P. M. 2013. Practical cross-docking optimization. In *TRISTAN VIII*. San Pedro de Atacama, Chile, 4–7.

[93] Guignard, M., Hahn, P. M., Pessoa, A. A., and da Silva, D. C. 2012. Algorithms for the cross-dock door assignment problem. In *Matheuristics'12 – Fourth International Workshop on Model Based Metaheuristics*. Rio de Janeiro, Brazil.

[94] Günther, M. and Nissen, V. 2010. Sub-daily staff scheduling for a logistics service provider. *Künstliche Intelligenz 24,* 2, 105–113.

[95] Günther, M. and Nissen, V. 2014. A comparison of three heuristics on a practical case of sub-daily staff scheduling. *Annals of Operations Research 218,* 1, 201–219.

[96] Guo, Y., Chen, Z.-R., Ruan, Y.-L., and Zhang, J. 2012. Application of NSGA-II with local search to multi-dock cross-docking scheduling problem. In *IEEE International Conference on Systems, Man, and Cybernetics*. Seoul, Korea.

[97] Guyon, O., Lemaire, P., Pinson, E., and Rivreau, D. 2010. Cut generation for an integrated employee timetabling and production scheduling problem. *European Journal of Operational Research 201,* 2, 557–567.

[98] Guyon, O., Lemaire, P., Pinson, E., and Rivreau, D. 2012. Solving an integrated job-shop problem with human resource constraints. *Annals of Operations Research 213,* 1, 147–171.

[99] Haneyah, S. W. A., Schutten, J. M. J., and Fikse, K. 2014. Throughput maximization of parcel sorter systems by scheduling inbound containers. In *Efficiency and Innovation in Logistics*, U. Clausen, M. ten Hompel, and J. F. Meier, Eds. Springer International Publishing, 147–159.

[100] Hauser, K. 2002. Simulation and optimization of a crossdocking operation in a just-in-time environment. Ph.D. thesis, University of Kentucky.

[101] Hazir, O., Haouari, M., and Erel, E. 2010. Robust scheduling and robustness measures for the discrete time/cost trade-off problem. *European Journal of Operational Research 207,* 2, 633–643.

[102] Hermel, D. Z. 2013. Three essays in operations management. Ph.D. thesis, University of British Columbia, Vancouver.

[103] Herroelen, W. and Leus, R. 2005. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research 165,* 2, 289–306.

[104] Jannat, S. and Greenwood, A. G. 2013. Using location parameters, moments, and percentiles to specify the triangular distribution. *Journal of Simulation*, 1–13.

[105] Jarrah, A. I., Qi, X., and Bard, J. F. 2014. The destination-loader-door assignment problem for automated package sorting centers. *Transportation Science*, 1–23.

[106] Ji, K. 2010. Truck scheduling problem at a cross-docking facility. M.S. thesis, University of Memphis.

[107] Joo, C. and Kim, B. 2013. Scheduling compound trucks in multi-door cross-docking terminals. *The International Journal of Advanced Manufacturing Technology 64,* 5-8, 977–988.

[108] Jurisica, I. 2000. Building better decision-support systems by using knowledge discovery. In *Proceedings of the 63rd ASIS Annual Meeting*. Chicago, USA, 281–291.

[109] Kapetanios, G., Vrisagotis, V., Pappas, D., Panta, M., and Siassiakos, K. 2009. A mathematical tool for warehousing optimization. In *WSEAS International Conference on Simulation, Modelling and Optimization*. Athens, Greece.

[110] Kinnear, E. 1997. Is there any magic in cross-docking? *Supply Chain Management: An International Journal 2*, 49–52.

[111] Ko, C. S., Lee, H. K., Choi, E. J., and Kim, T. 2008. A genetic algorithm approach to dock door assignment in automated cross-docking terminal with restricted layout. In *International Conference on Genetic and Evolutionary Methods*. Las Vegas, Nevada, USA.

[112] Konur, D. and Golias, M. M. 2013a. Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty. *Computers & Industrial Engineering 65,* 4, 663–672.

[113] Konur, D. and Golias, M. M. 2013b. Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. *Transportation Research Part E: Logistics and Transportation Review 49,* 1, 71–91.

[114] Kouvelis, P. and Yu, G. 1997. *Robust discrete optimization and its applications*. Nonconvex Optimization and Its Applications, vol. 14. Springer Science & Business Media.

[115] Larbi, R., Alpan, G., Baptiste, P., and Penz, B. 2011. Scheduling cross docking operations under full, partial and no information on inbound arrivals. *Computers & Operations Research 38,* 6, 889–900.

[116] Larbi, R., Alpan, G., and Penz, B. 2009. Scheduling transshipment operations in a multiple inbound and outbound door crossdock. In *Computers & Industrial Engineering*. 227–232.

[117] Leon, V. J., Wu, S. D., and Storer, R. H. 1994. Robustness measures and robust scheduling for job shops. *IIE Transactions 26,* 5, 32–43.

[118] Lequy, Q., Bouchard, M., Desaulniers, G., Soumis, F., and Tachefine, B. 2012. Assigning multiple activities to work shifts. *Journal of Scheduling 15,* 2, 239–251.

[119] Lequy, Q., Desaulniers, G., and Solomon, M. M. 2012. A two-stage heuristic for multi-activity and task assignment to work shifts. *Computers & Industrial Engineering 63,* 4, 831–841.

[120] Ley, S. and Elfayoumy, S. 2007. Cross dock scheduling using genetic algorithms. In *International Symposium on Computational Intelligence in Robotics and Automation*. Jacksonville, Florida, USA.

[121] Li, K. and Chen, Y.-X. 2007. Applying critical chain in project scheduling and estimating buffer size based on fuzzy technique. In *IEEE International Conference on Industrial Engineering and Engineering Management*. Singapore, 1068–1072.

[122] Li, Y., Lim, A., and Rodrigues, B. 2004. Crossdocking – JIT scheduling with time windows. *Journal of the Operational Research Society 55,* 12, 1342–1351.

[123] Li, Z., He, W., Sim, C. H., and Chen, C. C. 2008. A solution for cross-docking operations planning, scheduling and coordination. In *IEEE International Conference on Service Operations and Logistics, and Informatics*. Beijing, China.

[124] Li, Z., Low, M. Y. H., and Lim, R. Y. G. 2009. Optimal decision-making on product allocation for crossdocking and warehousing operations. *International Journal of Services Operations and Informatics 4*, 352–365.

[125] Li, Z., Low, M. Y. H., Lim, R. Y. G., and Ma, B. 2008. Optimal decision-making on product ranking for crossdocking/warehousing operations. In *IEEE International Conference on Industrial Informatics*. Daejon, Korea.

[126] Li, Z., Low, M. Y. H., Shakeri, M., and Lim, R. Y. G. 2009. Cross-docking planning and scheduling: problems and algorithms. Tech. rep., Singapore Institute of Manufacturing Technology.

[127] Li, Z., Sim, C. H., He, W., and Chen, C. C. 2012. A solution for cross-docking operations planning, scheduling and coordination. *Journal of Service Science and Management 05,* 02, 111–117.

[128] Li, Z., Sim, C. H., Low, M. Y. H., and Lim, R. Y. G. 2008. Optimal product allocation for crossdocking and warehousing operations in FMCG supply chain. In *IEEE International Conference on Service Operations and Logistics and Informatics*. Beijing, China, 2963–2968.

[129] Liao, T. W., Egbelu, P. J., and Chang, P.-C. 2012. Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. *Applied Soft Computing 12,* 11, 3683–3697.

[130] Liao, T. W., Egbelu, P. J., and Chang, P.-C. 2013. Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations. *International Journal of Production Economics 141,* 1, 212–229.

[131] Lim, A., Ma, H., and Miao, Z. 2006a. Truck dock assignment problem with operational time constraint within crossdocks. In *Advances in Applied Artificial Intelligence*, M. Ali and R. Dapoigny, Eds. Lecture Notes in Computer Science, vol. 4031. Springer Berlin Heidelberg, 262–271.

[132] Lim, A., Ma, H., and Miao, Z. 2006b. Truck dock assignment problem with time windows and capacity constraint in transshipment network through crossdocks. In *Computational Science and Applications*, M. Gavrilova, O. Gervasi, V. Kumar, C. Tan, D. Taniar, A. Laganá, Y. Mun, and H. Choo, Eds. Lecture Notes in Computer Science, vol. 3982. Springer Berlin Heidelberg, 688–697.

[133] Liu, Y. and Takakuwa, S. 2009. Simulation-based personnel planning for materials handling at a cross-docking center under retail distribution environment. In *Proceedings of the Winter Simulation Conference*. Austin, TX, USA, 2414–2425.

[134] Madani-Isfahani, M., Tavakkoli-Moghaddam, R., and Naderi, B. 2014. Multiple cross-docks scheduling using two meta-heuristic algorithms. *Computers & Industrial Engineering 74*, 129–138.

[135] Maknoon, M., Kone, O., and Baptiste, P. 2014. A sequential priority-based heuristic for scheduling material handling in a satellite cross-dock. *Computers & Industrial Engineering 72*, 43–49.

[136] Maknoon, M. Y. and Baptiste, P. 2009. Cross-docking: increasing platform efficiency by sequencing incoming and outgoing semi-trailers. *International Journal of Logistics Research and Applications 12,* 4, 249–261.

[137] Maknoon, M. Y. and Baptiste, P. 2010. Moving freight inside cross docking terminals. In *International Conference on Supply Chain Management and Information Systems*. Hong Kong.

[138] Maknoon, M. Y., Baptiste, P., and Koné, O. 2009. Optimal loading and unloading policy in cross-docking platform. In *Information Control Problems in Manufacturing*. Moscow, Russia, 1280–1285.

[139] Maravelias, C. T. and Sung, C. 2009. Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers & Chemical Engineering 33,* 12, 1919–1930.

[140] McWilliams, D. L. 2005. Simulation-based scheduling for parcel consolidation terminals: a comparison of iterative improvement and simulated annealing. In *Proceedings of the Winter Simulation Conference*. Orlando, FL, USA, 2087–2093.

[141] McWilliams, D. L. 2009a. A dynamic load-balancing scheme for the parcel hub-scheduling problem. *Computers & Industrial Engineering 57,* 3, 958–962.

[142] McWilliams, D. L. 2009b. Genetic-based scheduling to solve the parcel hub scheduling problem. *Computers & Industrial Engineering 56,* 4, 1607–1616.

[143] McWilliams, D. L. 2010. Iterative improvement to solve the parcel hub scheduling problem. *Computers & Industrial Engineering 59,* 1, 136–144.

[144] McWilliams, D. L. and McBride, M. E. 2012. A beam search heuristics to solve the parcel hub scheduling problem. *Computers & Industrial Engineering 62,* 4 (May), 1080–1092.

[145] McWilliams, D. L., Stanfield, P. M., and Geiger, C. D. 2005. The parcel hub scheduling problem: A simulation-based solution approach. *Computers & Industrial Engineering 49,* 3, 393–412.

[146] McWilliams, D. L., Stanfield, P. M., and Geiger, C. D. 2008. Minimizing the completion time of the transfer operations in a central parcel consolidation terminal with unequal-batch-size inbound trailers. *Computers & Industrial Engineering 54,* 4, 709–720.

[147] Mehta, S. and Uzsoy, R. 1998. Predictable scheduling of a job shop subject to breakdowns. *IEEE Transactions on Robotics and Automation 14,* 3, 365–378.

[148] Mercier, A. and Soumis, F. 2007. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research 34,* 8, 2251–2265.

[149] Miao, Z., Lim, A., and Ma, H. 2009. Truck dock assignment problem with operational time constraint within crossdocks. *European Journal of Operational Research 192,* 1, 105–115.

[150] Min, H. 2006. The applications of warehouse management systems: an exploratory study. *International Journal of Logistics Research and Applications 9,* 2, 111–126.

[151] Muller, P., Fayard, O., and Hiebel, P. 2013. Le Lean appliqué à l'ingénierie : un Graal inaccessible ? *Les Echos* 19/11/2013.

[152] Musliu, N., Gärtner, J., and Slany, W. 2002. Efficient generation of rotating workforce schedules. *Discrete Applied Mathematics 118,* 1-2, 85–98.

[153] Napolitano, M. 2000. *Making the move to cross docking: A practical guide to planning, designing and implementing a cross dock operation*. Warehousing Education & Research Council.

[154] Naudin, E., Chan, P. Y. C., Hiroux, M., Zemmouri, T., and Weil, G. 2012. Analysis of three mathematical models of the Staff Rostering Problem. *Journal of Scheduling 15,* 1, 23–38.

[155] Naujoks, B. and Chmielewski, A. 2009. Comparing different approaches on the door assignment problem in LTL-terminals. In *Models and Algorithms for Optimization in Logistics*, C. Barnhart, U. Clausen, U. Lauther, and R. H. Möhring, Eds.

[156] Nikulin, Y. 2006. Robustness in combinatorial optimization and scheduling theory: an extended annotated bibliography. Tech. rep., Universität Kiel.

[157] Nourmohammasi Sharabiani, S. 2009. Obtaining optimal and approximate solutions to the problem of scheduling inbound and outbound trucks in cross docking operations. M.S. thesis, University of Borås, School of Engineering.

[158] Oh, Y., Hwang, H., Cha, C. N., and Lee, S. 2006. A dock-door assignment problem for the Korean mail distribution center. *Computers & Industrial Engineering 51,* 2, 288–296.

[159] Peck, K. E. 1983. Operational analysis of freight terminals handling less than container load shipments. Ph.D. thesis, University of Illinois at Urbana-Champaign.

[160] Polge, C. 2014. "Des usines à colis" et au-delà ! *Supply Chain Magazine* 84 (May), p5.

[161] Porter, M. E. 1980. *Competitive strategy: Techniques for analyzing industries and competitors*. FreePress, New York.

[162] Pöyhönen, M. and Hämäläinen, R. P. 2001. On the convergence of multiattribute weighting methods. *European Journal of Operational Research 129,* 3, 569–585.

[163] Project Management Institute. 2008. *Guide to the project management body of knowledge*, 4th ed.

[164] PwC. 2013. Next-generation supply chains: Efficient, fast and tailored. Tech. rep.

[165] Qiu, Q., Zhang, Z., Song, X., and Gui, S. 2009. Application research of cross docking logistics in food cold-chain logistics. In *International Conference on Information Management, Innovation Management and Industrial Engineering*. 236–240.

[166] Rognon, J.-L. 2009. Carrefour: Des CCC pour tendre les flux de produits secs. *Supply Chain Magazine* 34 (May), 22–24.

[167] Rohrer, M. 1995. Simulation and cross-docking. In *Proceedings of the Winter Simulation Conference*. Arlington, VA, USA, 846–849.

[168] Rosales, C. R., Fry, M. J., and Radhakrishnan, R. 2009. Transfreight reduces costs and balances workload at Georgetown crossdock. *Interfaces 39,* 4, 316–328.

[169] Ross, J. 1997. Office Depot scores major cross-docking gains. *Store Mag*, 39.

[170] Sabuncuoglu, I. and Goren, S. 2009. Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *International Journal of Computer Integrated Manufacturing 22,* 2, 138–157.

[171] Saddle Creek Corporation. 2011. 2011 cross-docking trends report.

[172] Sadykov, R. 2009. A polynomial algorithm for a simple scheduling problem at cross docking terminals. Tech. Rep. RR-7054, INRIA.

[173] Sadykov, R. 2012. Scheduling incoming and outgoing trucks at cross docking terminals to minimize the storage cost. *Annals of Operations Research 201,* 1, 423–440.

[174] Saharidis, G. K. D., Golias, M. M., and Zhang, T. 2012. Discrete time formulation for the assignment problem applied in cross docking facilities. In *Transportation Research Board 91st Annual Meeting*. Washington DC, USA.

[175] Sandal, S. 2005. Staging approaches to reduce overall cost in a cross-dock environment. Ph.D. thesis, University of Missouri-Columbia.

[176] Schaerf, A. and Meisels, A. 2000. Solving employee timetabling problems by generalized local search. In *Advances in Artificial Intelligence*. Lecture Notes in Computer Science, vol. 1792. Springer Berlin Heidelberg, 380–389.

[177] Service de l'observation et des Statistiques. 2012. Les entrepôts et leur activité en 2010. Tech. rep., Commissariat général au développement durable.

[178] Shakeri, M. 2011. Truck scheduling problem in logistics of crossdocking. Tech. rep., Nanyang Technological University, School of Computer Engineering, Singapore.

[179] Shakeri, M. and Low, M. Y. H. 2010. Development of a novel resource-constrained crossdocking model for the truck scheduling problem. In *IEEE Conference on Emerging Technologies & Factory Automation*. Bilbao, Spain.

[180] Shakeri, M., Low, M. Y. H., and Li, Z. 2008. A generic model for crossdock truck scheduling and truck-to-door assignment problems. In *IEEE International Conference on Industrial Informatics*. Daejeon, Korea, 857–864.

[181] Shakeri, M., Low, M. Y. H., Li, Z., and Lee, E. W. 2010. Two efficient constructive heuristics for scheduling trucks at crossdocking terminals. In *IEEE International Conference on Service Operations and Logistics and Informatics*. QingDao, China, 177–182.

[182] Shakeri, M., Low, M. Y. H., Turner, S. J., and Lee, E. W. 2012. A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. *Computers & Operations Research 39,* 11, 2564–2577.

[183] Shiguemoto, A. L., Cavalcante Netto, U. S., and Bauab, G. H. S. 2014. An efficient hybrid meta-heuristic for a cross-docking system with temporary storage. *International Journal of Production Research 52,* 4, 1231–1239.

[184] Smet, P. and Vanden Berghe, G. 2012. A matheuristic approach to the shift minimisation personnel task scheduling problem. In *International Conference on the Practice and Theory of Automated Timetabling*. Son, Norway, 145–157.

[185] Soltani, R. and Sadjadi, S. J. 2010. Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. *Transportation Research Part E: Logistics and Transportation Review 46,* 5, 650–666.

[186] Song, K. and Chen, F. 2007. Scheduling cross docking logistics optimization problem with multiple inbound vehicles and one outbound vehicle. In *IEEE International Conference on Automation and Logistics*. Jinan, China, 3089–3094.

[187] Stalk, G., Evans, P., and Shulman, L. E. 1992. Competing on capabilities: the new rules of corporate strategy. *Harvard Business Review 63,* 2, 57–69.

[188] Stefanini, L., Sorini, L., and Guerra, M. L. 2008. Fuzzy numbers and fuzzy arithmetic. In *Handbook of Granular Computing*, W. Pedrycz, A. Skowron, and V. Kreinovich, Eds. John Wiley & Sons, Chapter 12, 249–284.

[189] Stickel, M. and Furmans, K. 2006. An optimal control policy for crossdocking terminals. In *Operations Research Proceedings*, H.-D. Haasis, H. Kopfer, and J. Schönberger, Eds. Springer Berlin Heidelberg, 79–84.

[190] Supply Chain Magazine. 2013. Principaux éditeurs de WMS - 7ème édition.

[191] Tang, S.-L. and Yan, H. Y. 2010. Pre-distribution vs. post-distribution for cross-docking with transshipments. *Omega 38,* 3-4, 192–202.

[192] Tesch, C., Clausen, U., and Wohlgemuth, S. 2011. Logistics for decision support in the operation of both terminals and the corresponding yards. In *15th World Multi-Conference on Systemics, Cybernetics and Informatics*. Orlando, FL, USA.

[193] Tsui, L. Y. and Chang, C.-H. 1990. A microcomputer based decision support tool for assigning dock doors in freight yards. *Computers & Industrial Engineering 19,* 1-4, 309–312.

[194] Tsui, L. Y. and Chang, C.-H. 1992. An optimal solution to a dock door assignment problem. *Computers & Industrial Engineering 23,* 1–4, 283–286.

[195] Vahdani, B., Soltani, R., and Zandieh, M. 2010. Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics. *The International Journal of Advanced Manufacturing Technology 46,* 5, 769–783.

[196] Vahdani, B., Tavakkoli-Moghaddam, R., and Mousavi, S. 2013. Scheduling of trucks in cross-docking systems: a hybrid meta-heuristic algorithm. In *International Conference on Applied Operational Research*. Lisbon, Portugal, 125–132.

[197] Vahdani, B. and Zandieh, M. 2010. Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Computers & Industrial Engineering 58,* 1, 12–24.

[198] Valckenaers, P., Folan, P., Vallespir, B., Ducq, Y., Cavalieri, S., Pfeiffer, A., Kádár, B., and Monostori, L. 2007. Stability-oriented evaluation of rescheduling strategies, by using simulation. *Computers in Industry 58,* 7, 630–643.

[199] Van Belle, J., Valckenaers, P., and Cattrysse, D. 2012. Cross-docking: State of the art. *Omega 40,* 6, 827–846.

[200] Van Belle, J., Valckenaers, P., Vanden Berghe, G., and Cattrysse, D. 2013. A tabu search approach to the truck scheduling problem with multiple docks and time windows. *Computers & Industrial Engineering 66,* 4, 818–826.

[201] van de Vonder, S., Demeulemeester, E., and Herroelen, W. 2008. Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research 189,* 3, 723–733.

[202] van de Vonder, S., Demeulemeester, E., Herroelen, W., and Leus, R. 2005. The use of buffers in project management: The trade-off between stability and makespan. *International Journal of Production Economics 97,* 2, 227–240.

[203] van den Berg, J. P. 2007. *Integral warehouse management: the next generation in transparency, collaboration and warehouse management systems*. Management Outlook, Utrecht, The Netherlands.

[204] van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. 2013. Personnel scheduling: A literature review. *European Journal of Operational Research 226,* 3, 367–385.

[205] van Veldhoven, S., Post, G., and van der Veen, E. 2012. Days off scheduling – A 2-phase approach to personnel rostering. In *International Conference on the Practice and Theory of Automated Timetabling*. Son, Norway, 381–384.

[206] Vasiljevic, D., Stepanovic, M., and Manojlovic, O. 2013. Cross docking implementation in distribution of food products. *Economics of Agriculture 60,* 1, 91–101.

[207] Vis, I. F. A. and de Koster, R. M. B. M. 2003. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research 147,* 1, 1–16.

[208] Vis, I. F. A. and Roodbergen, K. J. 2004. Short term storage of goods in cross-docking operations. Tech. rep., Vrije Universiteit Amsterdam.

[209] Vis, I. F. A. and Roodbergen, K. J. 2008. Positioning of goods in a cross-docking environment. *Computers & Industrial Engineering 54,* 3, 677–689.

[210] Vogt, J. J. 2010. The successful cross-dock based supply chain. *Journal of Business Logistics 31,* 1, 99–119.

[211] Wan, L. 2005. Staff planning and scheduling in the service industry: an application to US Postal Service mail processing and distribution centers. Ph.D. thesis, University of Texas.

[212] Wang, J.-F. and Regan, A. 2008. Real-time trailer scheduling for cross-dock operations. *Transportation Journal 57,* 3, 5–20.

[213] Weide, O., Ryan, D., and Ehrgott, M. 2010. An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research 37,* 5, 833–844.

[214] Werners, B. and Wülfing, T. 2010. Robust optimization of internal transports at a parcel sorting center operated by Deutsche Post World Net. *European Journal of Operational Research 201,* 2, 419–426.

[215] Wren, A. 1996. Scheduling, timetabling and rostering – A special relationship? In *International Conference on the Practice and Theory of Automated Timetabling*. Lecture Notes in Computer Science, vol. 1153. Edinburgh, UK, 46–75.

[216] Wu, Y., Dong, M., and Yang, D. 2011. Cross-docking centre operation optimization using simulation-based genetic algorithm. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 225,* 7, 1175–1187.

[217] Yan, H. Y. and Tang, S.-l. 2009. Pre-distribution and post-distribution cross-docking operations. *Transportation Research Part E: Logistics and Transportation Review 45,* 6, 843–859.

[218] Yu, V. F., Sharma, D., and Murty, K. G. 2008. Door allocations to origins and destinations at less-than-truckload trucking terminals. *Journal of Industrial and Systems Engineering 2,* 1, 1–15.

[219] Yu, W. 2002. Operational strategies for cross docking systems. Ph.D. thesis, Iowa State University.

[220] Yu, W. and Egbelu, P. J. 2008. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research 184,* 1, 377–396.

[221] Zhang, T. 2009. Inbound and outbound truck scheduling at crossdock. M.S. thesis, State University of New Jersey.

[222] Zhang, T., Saharidis, G. K. D., Theofanis, S., and Boile, M. 2010. Scheduling of inbound and outbound trucks at cross-docks. *Transportation Research Record: Journal of the Transportation Research Board 2162*, 9–16.

[223] Zhu, Y.-R., Hahn, P. M., Liu, Y., and Guignard, M. 2009. New approach for the cross-dock door assignment problem. In *XLI Brazilian Symposium on Operations Research*. Porto Seguro, Brazil.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/

RÉSUMÉ

Dans une plateforme de cross-docking, les produits sont déchargés des camions entrants, triés puis directement rechargés dans les camions sortants – chaque produit passe moins de 24 heures sur la plateforme.

L'analyse des écarts entre la littérature et les observations réalisées sur le terrain permet de dégager deux axes de recherche : la prise en compte des incertitudes opérationnelles d'une part, et de la capacité des ressources humaines de la plateforme d'autre part.

Le problème de planification des camions entrants et sortants avec fenêtre de temps est modélisé par un programme linéaire et résolu par trois heuristiques différentes. La robustesse des plannings obtenus est ensuite testée à l'aide d'un modèle de simulation à événements discrets, qui permet d'évaluer plusieurs reformulations robustes du modèle initial.

Le problème de planification des employés sur la plateforme est traité à l'aide de trois programmes linéaires mixtes, résolus de façon séquentielle. La combinaison des deux modèles permet d'obtenir un modèle d'aide à la décision pour une plateforme de cross-docking.

MOTS-CLÉS *Logistique, cross-docking, programmation linéaire, heuristiques, planification des camions, emplois du temps.*

ABSTRACT

In a cross-docking platform, goods are unloaded from inbound trucks, sorted and directly reloaded in outbound trucks – each product typically stays less than 24 hours in the platform.

By analyzing the gaps between the literature and on-field observations, we highlight two research directions: accounting for operational uncertainties, and for the human resource capacity in the platform.

A truck scheduling problem with time windows for the inbound and outbound trucks is modeled with an integer program and solved with three different heuristics. The robustness of the schedules obtained is then tested with a discrete-event simulation model, which enables to evaluate several robust reformulations of the initial model.

The employee timetabling and rostering problem in the platform is addressed with three mixed integer linear problems solved sequentially. The two models can be combined to serve as a decision-support tool for a cross-docking platform.

KEY WORDS *Logistics, cross-docking, linear programming, heuristics, truck scheduling, employee timetabling.*